

Windows Patchday

KIS Phoenix - JBoss Patch

24.11.2023

Management Summary

blabla

Inhaltsverzeichnis

Abkürzungen	3
1 Einführung	1
2 Generell	4
3 Testsystem	9
4 Produktivsystem	12
Abbildungsverzeichnis	15
Tabellenverzeichnis	16
Listings	17
Literatur	18
Glossar	19
Anhang	20
0.0.1 #811534244# - KIS PROD Down - SKS0021 Defekt	21

Abkürzungen

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

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.

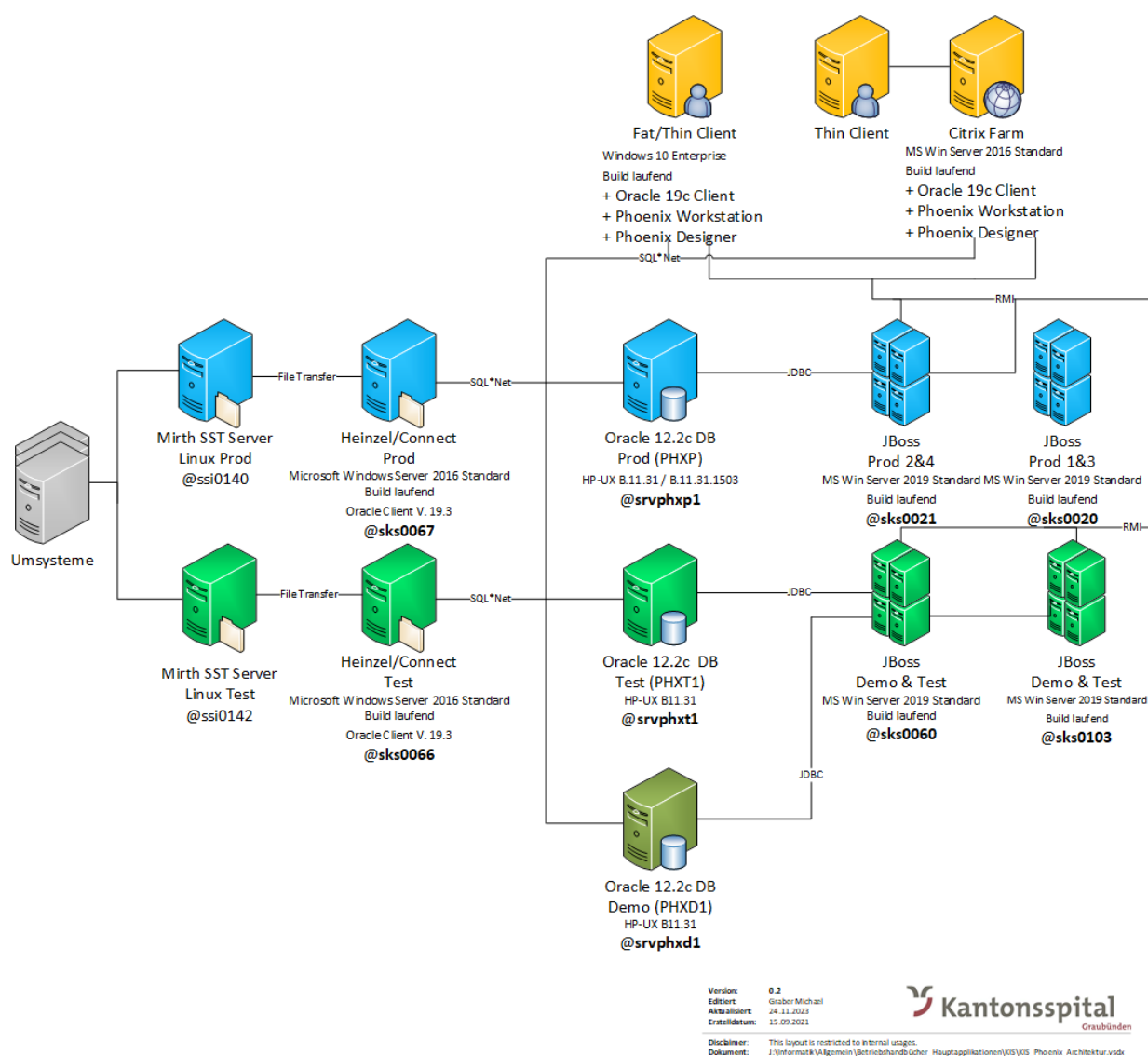


Abbildung 1.1: Architektur KIS Phoenix[5]

Das Test1 und Demosystem hat je einen JBoss-Node auf einer Seite, dass 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 Log-Rotation der Logs beträgt 12 Stunden. Das heisst, immer 12 Uhr Mittags und Mitternachts werden die Logs erneuert.

Bei diesem Schritt werden aber nicht nur die Logs neu erstellt, sondern auch der ganze Cache neu geladen. Der Cache wird zudem auch neu geladen, wenn ein Service rebooted wird, weshalb dieser Vorgang recht viel Zeit in Anspruch nimmt.

Die Nodes führen untereinander eine Art Load Balancing durch. Fällt ein Node oder Server aus, werden die User auf die anderen Nodes verteilt.

Das gilt auch innerhalb eines Servers, stoppe ich einen Node wird, wenn überhaupt möglich, dessen Last auf den zweiten Node des Servers gelegt.

Allerdings hat nur das Produktivsystem diese Konfiguration.

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: [Unterabschnitt 0.0.1](#)

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

Erkennen, ob ein JBoss-Service gestoppt ist, tut man daran, dass folgender Eintrag zuletzt im *server.log* steht:

```
1 2024-01-05 10:56:10,422 INFO [--] [org.jboss.as] ip[] mib[] user[] session[]
  request[] thread[MSC service thread 1-3]: WFLYSRV0050: JBoss EAP 7.4.8.GA (
  WildFly Core 15.0.19.Final-redhat-00001) stopped in 1199ms
2
```

Listing 2.2: Service gestoppt - server.log

Ein Erfolgreicher start erkennt man wiederum an folgenden Eintrag:

```
1 2024-01-05 11:01:36,000 INFO [--] [com.cgm.phoenix.common.masterdata.srv.catalog.
  dao.CatalogValueDAO] ip[] mib[] user[] session[] request[] thread[
  ServerService Thread Pool -- 273]: Creating folder index...
2 2024-01-05 11:01:36,047 INFO [--] [com.cgm.phoenix.common.masterdata.srv.catalog.
  dao.CatalogValueDAO] ip[] mib[] user[] session[] request[] thread[
  ServerService Thread Pool -- 273]: Creating search index...
3 2024-01-05 11:02:00,656 INFO [--] [com.cgm.phoenix.common.masterdata.srv.catalog.
  dao.CatalogValueDAO] ip[] mib[] user[] session[] request[] thread[
  ServerService Thread Pool -- 273]: Catalog data loaded.
4 2024-01-05 11:02:00,656 INFO [--] [com.cgm.phoenix.common.masterdata.srv.catalog.
  CatalogCacheProvider] ip[] mib[] user[] session[] request[] thread[
  ServerService Thread Pool -- 273]: Instant Catalogs successfully initialized.
5 2024-01-05 11:02:00,891 INFO [--] [org.jboss.as.server] ip[] mib[] user[] session
  [] request[] thread[ServerService Thread Pool -- 43]: WFLYSRV0010: Deployed "
  phoenix.ear" (runtime-name : "phoenix.ear")
```

```

6 2024-01-05 11:02:00,891 INFO  [--] [org.jboss.as.server] ip[] mib[] user[] session
    [] request[] thread[ServerService Thread Pool -- 43]: WFLYSRV0010: Deployed "
    jolokia.war" (runtime-name : "jolokia.war")
7 2024-01-05 11:02:00,922 INFO  [--] [org.jboss.as.server] ip[] mib[] user[] session
    [] request[] thread[Controller Boot Thread]: WFLYSRV0212: Resuming server
8 2024-01-05 11:02:01,000 INFO  [--] [org.jboss.as] ip[] mib[] user[] session[]
    request[] thread[Controller Boot Thread]: WFLYSRV0025: JBoss EAP 7.4.8.GA (
    WildFly Core 15.0.19.Final-redhat-00001) started in 281140ms - Started 18250
    of 18473 services (563 services are lazy, passive or on-demand)
9 2024-01-05 11:02:01,000 INFO  [--] [org.jboss.as] ip[] mib[] user[] session[]
    request[] thread[Controller Boot Thread]: WFLYSRV0060: Http management
    interface listening on http://10.0.22.52:9990/management
10 2024-01-05 11:02:01,016 INFO  [--] [org.jboss.as] ip[] mib[] user[] session[]
    request[] thread[Controller Boot Thread]: WFLYSRV0051: Admin console listening
    on http://10.0.22.52:9990
11

```

Listing 2.3: Service gestartet - server.log

Es kann aber mehrere Minuten dauern, bis ein Service vollständig gestartet ist.

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!

Obwohl die Server mittels Ivanti betankt werden, kann es vorkommen, dass nicht alle Patches geladen wurden. Um sicherzustellen, dass das System komplett sauber gepatched wurde, sollte man immer noch auf Updates Prüfen.

Geprüft wird Normal via Suche:

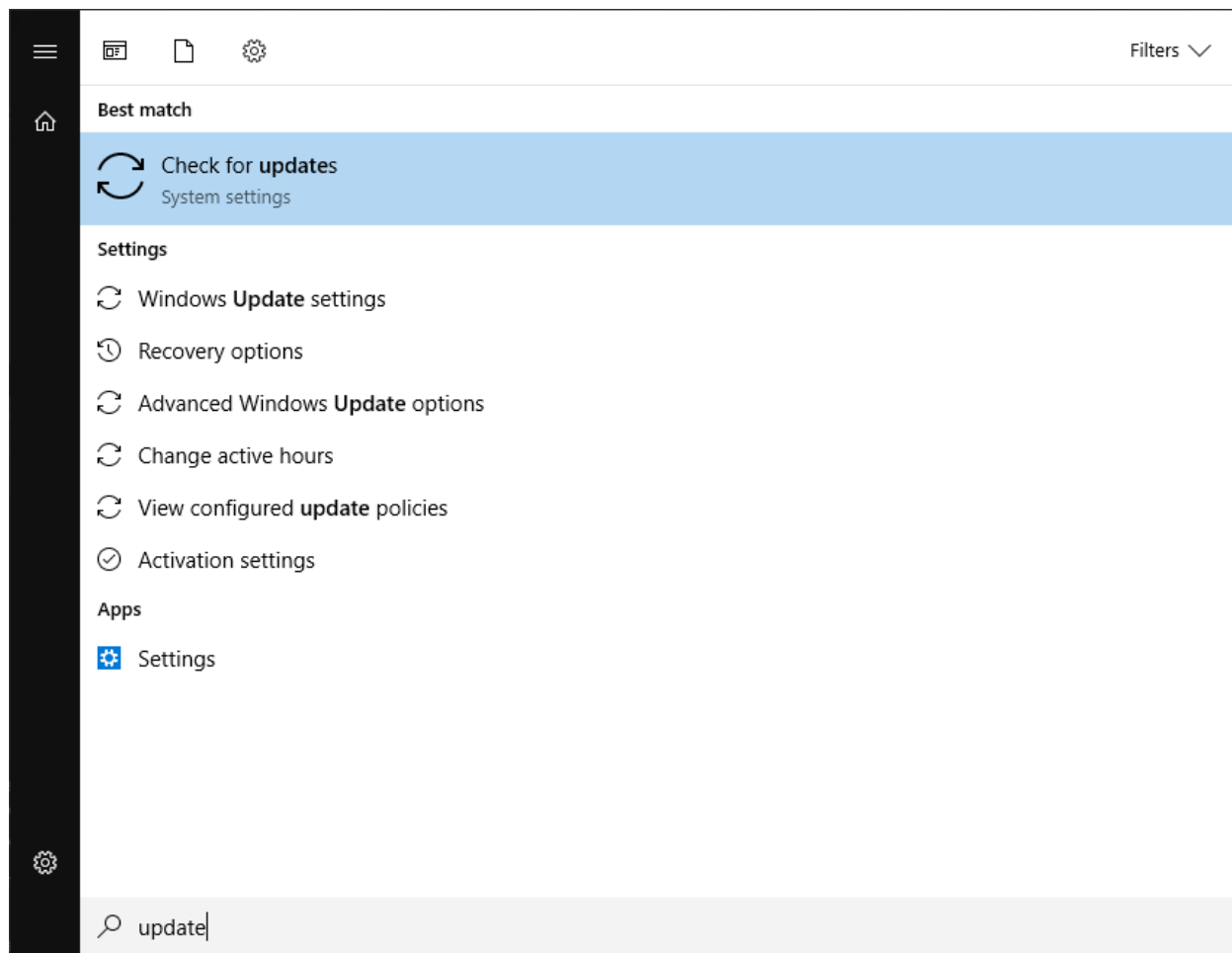


Abbildung 2.2: Check-Updates

Solange noch Updates wie folgt vorhanden sind, müssen diese manuell nachgeladen und installiert werden:

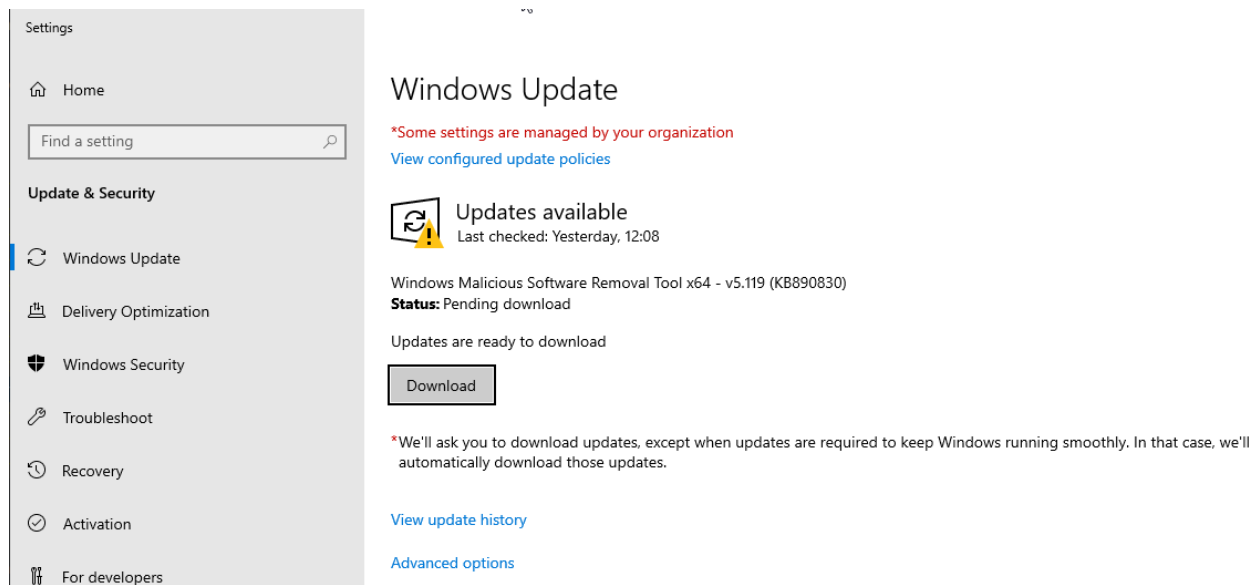


Abbildung 2.3: Updates available

Dies ist aber äusserst selten der Fall.

Vorsicht!

Sollte es trotzdem mal vorkommen, niemals die RDP-Verbindung schliessen!
Ansonsten wird es unmöglich sein, den Server gezielt zu rebooten.

Desweiteren muss auf der VMware vSphere die Berechtigung gegeben sein, Snapshots erstellen und löschen zu können. Auch Restores via Snapshot sollten machbar sein.

3 Testsystem

Bevor der Patch eingespielt werden kann, müssen die Voraussetzungen erfüllt sein. Da das Testsystem im laufenden Betrieb gepatched wird und Standardmässig die Workstation.exe für das Testsystem nur auf dem Laufwerk K:\ steht, muss ein Lokales verzeichnis im C:\ erstellt werden.

Ausserdem sollten zwei Ini-Files präperiert werden.

sks0060

```
1 -clean
2 -data
3 @noDefault
4 -configuration
5 C:\temp\phoenix\test1_7_23_1\eclipse\configuration
6 -nl
7 de_CH
8 -vmargs
9 -Dorg.eclipse.update.reconcile=false
10 -Dphoenix.db=Phoenix_Test1
11 -Dphoenix.application.workspace.createdir=true
12 -Dphoenix.application.workspace=\\.\phoenix\workspace\test1_7_23_1
13 -Dphoenix.login.usernamefile=%APPDATA%/Phoenix/login_username_test1
14 -Dphoenix.server.nodes=sks0060:8443
15 -Djavax.net.ssl.trustStore=.\truststore.jks
16 -Dphoenix.login.enableLDAP=true
17 -Dphoenix.application.winlogin=true
18 -Dphoenix.g3his.url=http://clinical-qa.ksgr.ch:8280/cgmg3
19
```

Listing 3.1: Workstation.ini TEST sks0060

sks0103

```
1 -clean
2 -data
3 @noDefault
4 -configuration
5 C:\temp\phoenix\test1_7_23_1\eclipse\configuration
6 -nl
7 de_CH
8 -vmargs
9 -Dorg.eclipse.update.reconcile=false
10 -Dphoenix.db=Phoenix_Test1
```

```

11 -Dphoenix.application.workspace.createdir=true
12 -Dphoenix.application.workspace=\\\\phoenix\\workspace\\test1_7_23_1
13 -Dphoenix.login.usernamefile=%APPDATA%/Phoenix/login_username_test1
14 -Dphoenix.server.nodes=sks0103:8443
15 -Djavax.net.ssl.trustStore=.\truststore.jks
16 -Dphoenix.login.enableLDAP=true
17 -Dphoenix.application.winlogin=true
18 -Dphoenix.g3his.url=http://clinical-qa.ksgr.ch:8280/cgmg3
19

```

Listing 3.2: Workstation.ini TEST sks0103

Vorsicht!

Bei jedem KIS Phoenix Test1 Update kann sich die Konfiguration ändern!
Daher sollten die Inis immer neu generiert werden bei den Updates.

Folgende Schritte müssen gemacht werden, um den Patch durchzuführen:

1. Snapshost der beiden Server sks0060 und sks0103 machen
2. Prüfen ob Workstation.exe Test1 auf sks0060 und sks0103 verbinden kann
3. Prüfen ob Workstation.exe Test1 nur auf sks0103 verbinden kann
4. Den Dienst PhoenixJBossEAP_test1_1 auf sks0060 stoppen
5. Server sks0060 rebooten
6. Auf sks0060 prüfen, ob keine Patches mehr gezogen werden könnten
7. Prüfen ob Workstation.exe Test1 auf sks0060 verbinden kann
8. Prüfen ob Workstation.exe Test1 auf sks0103 verbinden kann
9. Prüfen ob Workstation.exe Test1 auf sks0060 und sks0103 verbinden kann
10. Den Dienst PhoenixJBossEAP_test1_2 auf sks0103 stoppen
11. Server sks0103 rebooten
12. Auf sks0103 prüfen, ob keine Patches mehr gezogen werden könnten
13. Prüfen ob Workstation.exe Test1 auf sks0103 verbinden kann
14. Prüfen ob Workstation.exe Test1 auf sks0060 verbinden kann
15. Prüfen ob Workstation.exe Test1 auf sks0060 und sks0103 verbinden kann
16. Wenn die Tests erfolgreich waren, Snapshost der beiden Server sks0060 und sks0103 löschen

Das ganze noch als Sequenzdiagramm dargestellt:

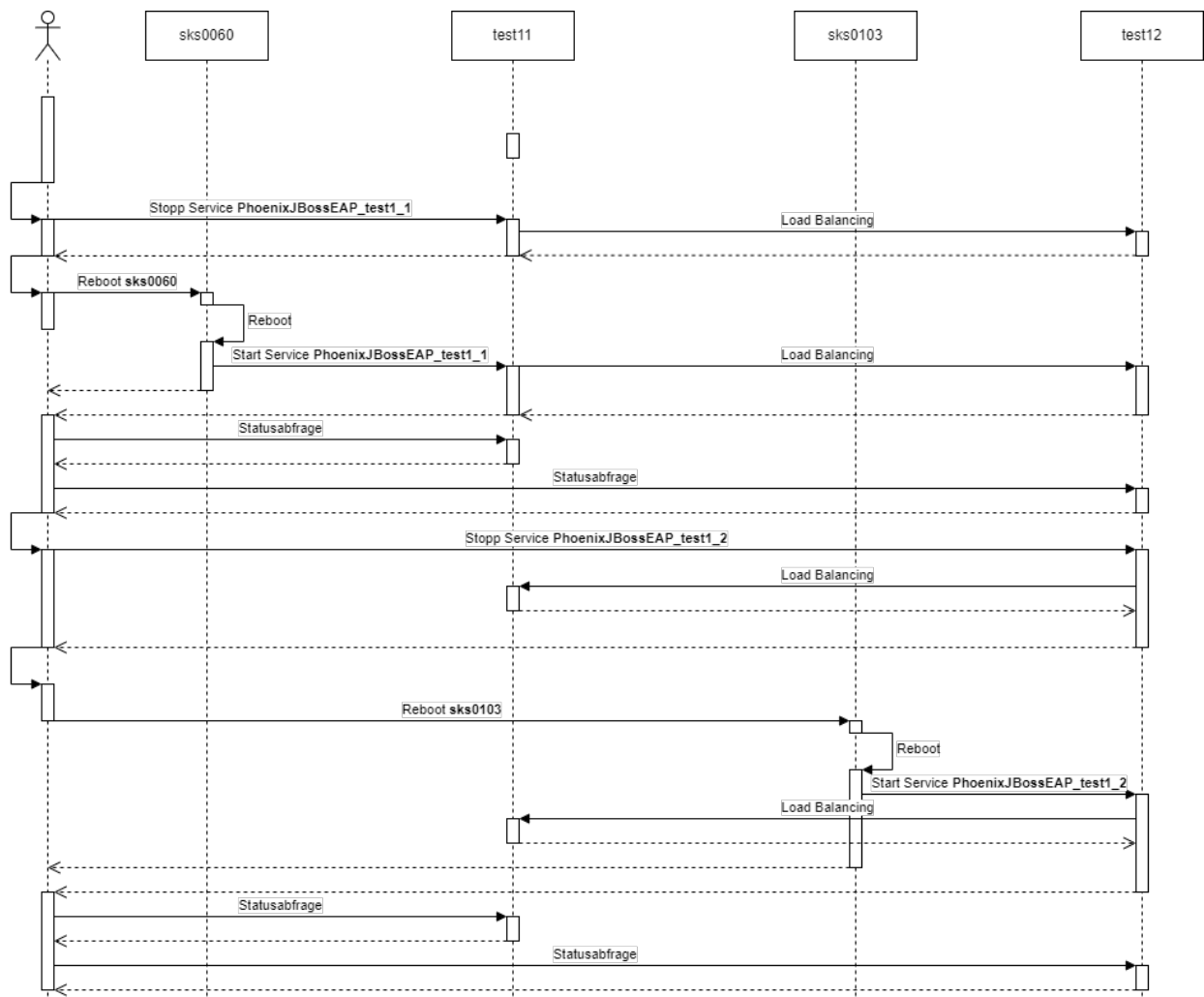


Abbildung 3.1: TEST-Sequenzdiagramm

4 Produktivsystem

Ausserdem sollten ebenfalls zwei Ini-Files präperiert werden.

sks0020

```
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,sks0020:8543
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 4.1: Workstation.ini PROD sks0020

sks0020

```
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=sks0021:8443,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
```


Listing 4.2: Workstation.ini PROD sks0021

Vorsicht!

Beim Installieren einer neuen produktiven Workstation.exe wird das ganze Verzeichnis jeweils auf dem Laufwerk C:\ überschrieben!

Daher müssen die Inis immer wieder neu erstellt werden.

Folgende Schritte müssen gemacht werden, um den Patch durchzuführen:

1. Snapshot der beiden Server sks0020 und sks0021 machen
2. Prüfen ob Workstation.exe Prod auf sks0020 und sks0021 verbinden kann
3. Prüfen ob Workstation.exe Prod nur auf sks0021 verbinden kann
4. Den Dienst PhoenixJBossEAP_prod_3 auf sks0020 stoppen
5. Prüfen ob Workstation.exe Prod nur auf sks0020 verbinden kann
6. Den Dienst PhoenixJBossEAP_prod_1 auf sks0020 stoppen
7. Server sks0020 rebooten
8. Auf sks0020 prüfen, ob keine Patches mehr gezogen werden könnten
9. Prüfen ob Workstation.exe Prod auf sks0020 verbinden kann
10. Prüfen ob Workstation.exe Prod auf sks0021 verbinden kann
11. Prüfen ob Workstation.exe Prod auf sks0020 und sks0021 verbinden kann
12. Den Dienst PhoenixJBossEAP_prod_4 auf sks0021 stoppen
13. Prüfen ob Workstation.exe Prod nur auf sks0021 verbinden kann
14. Den Dienst PhoenixJBossEAP_prod_2 auf sks0021 stoppen
15. Server sks0021 rebooten
16. Auf sks0021 prüfen, ob keine Patches mehr gezogen werden könnten
17. Prüfen ob Workstation.exe Prod auf sks0021 verbinden kann
18. Prüfen ob Workstation.exe Prod auf sks0020 verbinden kann
19. Prüfen ob Workstation.exe Prod auf sks0020 und sks0021 verbinden kann

20. Nach einem Tag ohne Fehler, die Snapshots der beiden Server sks0020 und sks0021 löschen

Das ganze noch als Sequenzdiagramm dargestellt:

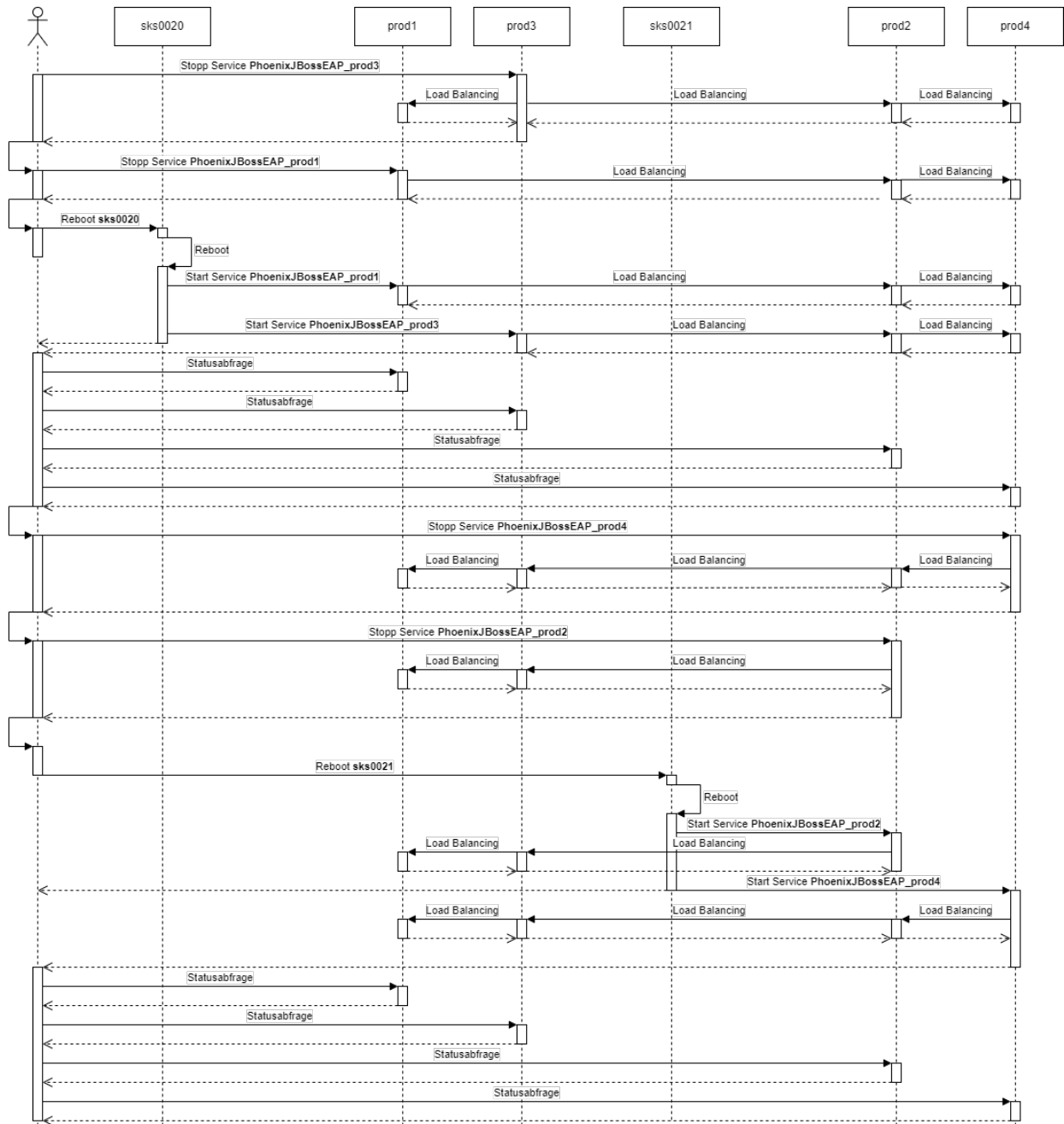


Abbildung 4.1: PROD-Sequenzdiagramm

Abbildungsverzeichnis

1.1	Architektur KIS Phoenix[5]	1
2.1	Deployment-Status	4
2.2	Check-Updates	7
2.3	Updates available	8
3.1	TEST-Sequenzdiagramm	11
4.1	PROD-Sequenzdiagramm	14
.1	#811534244# - KIS PROD Down - SKS0021 Defekt	21

Tabellenverzeichnis

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

Listings

2.1	Workstation.ini PROD Beispiel	5
2.2	Service gestoppt - server.log	5
2.3	Service gestartet - server.log	5
3.1	Workstation.ini TEST sks0060	9
3.2	Workstation.ini TEST sks0103	9
4.1	Workstation.ini PROD sks0020	12
4.2	Workstation.ini PROD sks0021	12

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] Cloudflare Germany GmbH. *Was ist das Remote Desktop Protocol (RDP)?* <https://www.cloudflare.com/de-de/learning/access-management/what-is-the-remote-desktop-protocol/>. 2024.
- [5] Michael Graber u. a. *KIS-Betriebshandbuch_V0_25.docx*. 2021.
- [6] Inc Ivanti. *AppSense - Ivanti*. <https://www.ivanti.com/de/company/history/appsense>.

Glossar

Ivanti Das KSGR nutzt die Ivanti AppSense[6, 2] für die Verteilung von Software und dem Installieren von Patches.. 1, 6

Java Java ist eine Hochsprache die 1995 von Sun Microsystems veröffentlicht wurde. Dank der JVM läuft Java auf sehr vielen Plattformen[3].. 3, 19

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

RDP Microsofts Protokoll für die Übertragung von Bildschirm- und Peripheriedaten von einem Remote-Rechner zu übertragen[4].. 8

VMware vSphere Die vSphere® ist ein Typ-1 Hypervisor der Firma VMware® der eine Reihe leistungsstarker Tools und Funktionen mitbringt.. 8

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.

