OC Pizza Company SB Design

# OC Pizza Company

# Information Management System

## Program Roll Out Procedure

Version 1.0

**Author**
Scott Bolin
*Engineer*

# TABLE OF CONTENTS

**1 - Versions** **3**
**2 - Introduction** **4**
2.1 - Document purpose 4
2.2 - References 4
**3 - Prerequisites** **5**
3.1 - System 5
*3.1.1 - Web Server* *5*
3.1.1.1 - Technical specifications 5
3.1.1.2 - Hardware requirements, 5
*3.1.2 - Application Server* *5*
3.1.2.1 - Technical specifications 5
3.1.2.2 - Hardware requirements, 5
*3.1.3 - Push Notification* *5*
3.1.3.1 - Technical specifications 5
*3.1.4 - Database Server* *6*
3.1.4.1 - Technical specifications 6
*3.1.5 - Authorization Framework* *6*
3.1.5.1 - Technical specifications 6
3.2 - Databases 6
3.3 - Web services 6
3.4 - Other Resources 6
**4 - Roll-out procedure** **7**
4.1 - Roll-Out Overview 7
4.2 - Roll-out of Web Server 7
*4.2.1 - Apache Tomcat* *7*
*4.2.2 - Environment variable* *8*
*4.2.3 - Key Directories* *8*
*4.2.4 - Verification* *8*
4.3 - Roll-out of Database 9
*4.3.1 - MySQL Install* *9*
*4.3.2 - Application configuration directory* *9*
*4.3.3 - Datasources* *9*
*4.3.4 - Testing* *9*
4.4 - Roll-out of Notification Server 10
*4.4.1 - Uniqush install* *10*
*4.4.2 - Application configuration directory* *11*
*4.4.3 - Testing* *12*
4.5 - Roll-out of Mobile Applications 12
*4.5.1 - Application Installation* *12*
**5 - Start/Shutdown procedure** **13**
5.1 - Webserver 13
5.2 - Database 13
**6 - Update procedure** **14**
6.1 - Webserver 14
6.2 - Database 15
**7 - Supervision/Monitoring** **16**
7.1 - Webserver 16
7.2 - Database 16
**8 - Backup and restore procedure** **17**

# 1 - Versions

| Author | Date | Description | Version |
|---|---|---|---|
| SB | 03/10/2020 | Creation of document | 1.00 |
| | | | |

# 2 - Introduction

## 2.1 - Document purpose

This document details the program roll out procedure for the OC Pizza Company Information Management System application.

The main objective is to detail the roll out procedures used in the start-up of the OC Pizza Company IMS back end database

## 2.2 - References

For further information, please refer to:

1. **FDF - 1**: Functional design for the application
2. **TDF – 1**: Technical Design File
3. **DN – 1**: Delivery Note File

# 3 - Prerequisites

## 3.1 - System

### 3.1.1 - Web Server

Apache Tomcat Server

#### 3.1.1.1 - Technical specifications

Tested OS and platforms

- FreeBSD 3 — 12 / i386; FreeBSD 5 — 12 / amd64; FreeBSD 11 / ppc; FreeBSD 12 / ppc64;
- Linux 2.2 — 4 / i386; Linux 2.6 — 5 / amd64; Linux 3 — 4 / armv6l, armv7l, aarch64, ppc64le;
- Solaris 9 / i386, sun4u; Solaris 10 / i386, amd64, sun4v; Solaris 11 / x86;
- AIX 7.1 / PowerPC;
- HP-UX 11.31 / ia64;
- macOS / PowerPC, i386, x86_64;
- Windows XP, Windows Server 2003, Windows 7, Windows 10.

#### 3.1.1.2 - Hardware requirements,

Intel/AMD CPU with at least 64GB RAM and a 1GB network interface is required to handle the anticipated network loads and to provide sufficient memory for any caching required.

### 3.1.2 - Application Server

JBOSS Application Server

#### 3.1.2.1 - Technical specifications

Tested OS and platforms

- FreeBSD 3 — 12 / i386; FreeBSD 5 — 12 / amd64; FreeBSD 11 / ppc; FreeBSD 12 / ppc64;
- Linux 2.2 — 4 / i386; Linux 2.6 — 5 / amd64; Linux 3 — 4 / armv6l, armv7l, aarch64, ppc64le;
- Solaris 9 / i386, sun4u; Solaris 10 / i386, amd64, sun4v; Solaris 11 / x86;
- AIX 7.1 / PowerPC;
- HP-UX 11.31 / ia64;
- macOS / PowerPC, i386, x86_64;
- Windows XP, Windows Server 2003, Windows 7, Windows 10.

#### 3.1.2.2 - Hardware requirements,

Intel/AMD multicore CPU (minimum 8 cores/16 threads, with more cores encouraged. At least 32GB RAM and a 1GB network interface is required to handle the anticipated network loads and to provide sufficient memory for any caching required.

### 3.1.3 - Push Notification

Uniqush provides a unified push service for server-side notification to apps on mobile devices. By running Uniqush on server side you can send push notification to any supported mobile platform.

#### 3.1.3.1 - Technical specifications

A typical install of Uniqush is as a Linux package; however, Uniqush is open source with source code available and can be compiled to different operating systems with some alterations.

Uniqush is dependent on Redis, an open source light weight database used for notification storage. The latest version upon time of writing is 6.0.8. Redis is also an open source project, with source code available. Source can be downloaded via wget and compiled with make on many platforms.

### *3.1.4 - Database Server*

Database server hosting MySQL v8 database schema/base(s)

#### *3.1.4.1 - Technical specifications*

For the online database, any modern (within last 2 year) computer running relatively recent version of FreeBSD, Linux, MacOS, or Windows OS can be used as a server hosting MySQL. At least 2TB storage is required, with EDSFF ("Edge") long or short form factor SSD highly recommended, with a minimum of PCIe 4.0 NVMe SSD. For safety, storage should be configured in a RAID setup. RAID 10 is preferred for redundancy, speed, and failover safety but does require 4 drives to be used.

### *3.1.5 - Authorization Framework*

Elytron framework is used to handle authentication.

#### *3.1.5.1 - Technical specifications*

The Elytron framework is part of the JBOSS Application server package and is used when authorization capabilities are required. For example, in the mobile applications, Elytron will handle user authentication..

## 3.2 - Databases

The following databases and schema must be accessible and up to date:

- MySQL version 8

## 3.3 - Web services

The following web services must accessible and up to date:

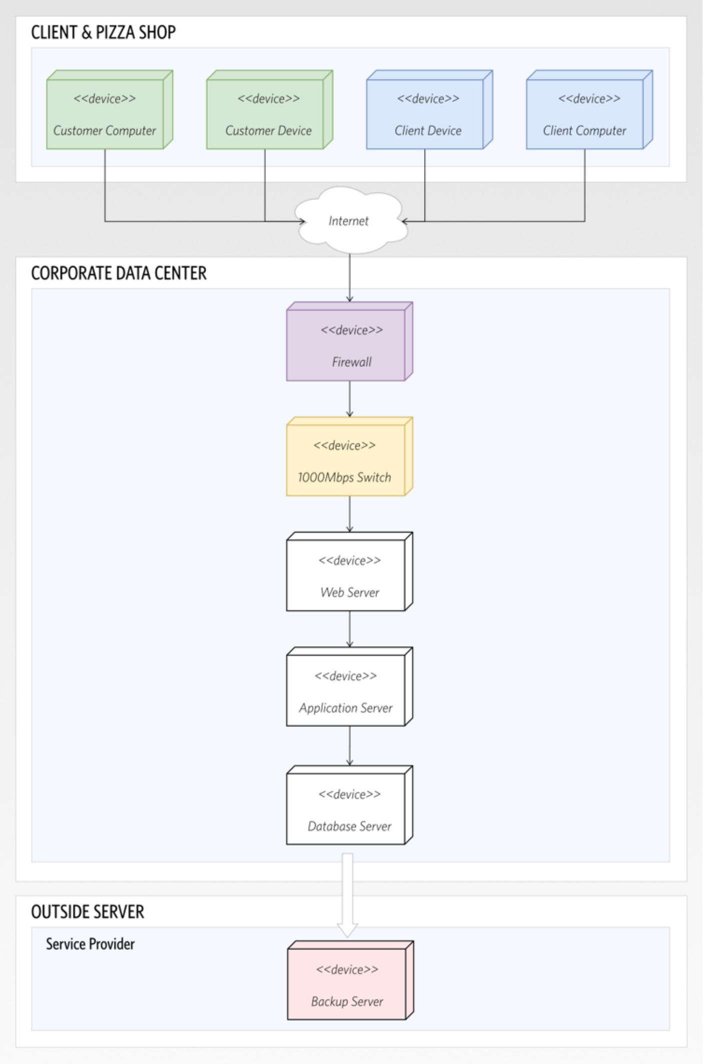
- REST services are used, which are automatically incorporated in the above Database/Web server systems;

## 3.4 - Other Resources

No other resources are used/needed.

# 4 - ROLL-OUT PROCEDURE

## 4.1 - Roll-Out Overview



## 4.2 - Roll-out of Web Server

### *4.2.1 - Apache Tomcat*

Basic install instructions:

1. **Download** and Install a Java SE Runtime Environment (JRE)
   a. Download a Java SE Runtime Environment (JRE), release version 8 or later, from: [JRE Download]
   b. Install the JRE according to the instructions included with the release.

You may also use a full Java Development Kit (JDK) rather than just a JRE.

2. **Download** and Install Apache Tomcat
   a. Download a binary distribution of Tomcat from: [Apache Tomcat]
   b. Unpack the binary distribution so that it resides in its own directory (conventionally named "apache-tomcat-[version]").

Full installation instructions can be found here: [Tomcat Setup].

### 4.2.2 - Environment variable

These are the environment variables recognized by the JBOSS application

| Name | Obligatory | Description |
|---|---|---|
| apache-tomcat-[ver] | Yes | Root directory for the installation of the application |

### 4.2.3 - Key Directories

Server:

/home/tomcat/apache-tomcat-[version]

Other important tomcat directories:

- **/bin** - Startup, shutdown, and other scripts. The *.sh files (for Unix systems) are functional duplicates of the *.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.
- **/conf** - Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.
- **/logs** - Log files are here by default.
- **/webapps** - This is where your webapps go.
- **/data/www** - Web pages
- **/images** – image file location for web pages
- **/data** – data files for web pages

### 4.2.4 - Verification

To verify Apache is running correctly, follow these steps:

1. Start the Apache Tomcat.

Switch to the Tomcat username in the target zone (in the following example, it is root) and change to the directory where the environment script is located. In the following example the Tomcat version is 4.1.24.

Note –

The output messages of the start and shutdown commands depend on version installed.

```
# . ./env.ksh
# cd $CATALINA_HOME/bin
# ./startup.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:       /usr/j2se
```

2. Check the Installation

Start a web browser and connect to the server with http://*nodename*:8080. If you see the default Tomcat homepage everything is working correctly.

3. Stop the Apache Tomcat.

In addition, review log for errors.

## 4.3 - Roll-out of Database

### *4.3.1 - MySQL Install*

MySQL comes with a stand-alone package installer for various Linux distributions, MacOS, and Windows, making installation relatively simple.

### *4.3.2 - Application configuration directory*

The directory structure depends on the Linux distribution used. For Debian, the application configuration directory must be created in the folder system and set as follows:

```
/usr/bin/mysql-version/
```

Templates:

```
/usr/bin/mysql-version/Templates/
```

packagerules.xml:

```
/etc/mysql/
```

Products.xml:

```
/etc/mysql/
```

Cache:

```
/usr/bin/mysql-version/cache/
```

### *4.3.3 - Datasources*

Typically, the data directory for MySQL is initialized automatically (with base install method). Accesses to the databases must be configured with folder

The data source is located at:

```
/var/lib/mysql
```

### *4.3.4 - Testing*

Use mysqladmin to verify that the server is running. The following commands provide simple tests to check whether the server is up and responding to connections:

```
shell> bin/mysqladmin version

shell> bin/mysqladmin variables
```

If you cannot connect to the server, specify a -u root option to connect as root. If you have assigned a password for the root account already, you'll also need to specify -p on the command line and enter the password when prompted. For example:

```
shell> bin/mysqladmin -u root -p version

Enter password: (enter root password here)
```

The output from **mysqladmin version** varies depending on your platform and version of MySQL, but should be similar to that shown here:

```
shell> bin/mysqladmin version
mysqladmin  Ver 14.12 Distrib 8.0.23, for pc-linux-gnu on i686
...
Server version          8.0.23
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 14 days 5 hours 5 min 21 sec
Threads: 1  Questions: 366  Slow queries: 0
Opens: 0  Flush tables: 1  Open tables: 19
Queries per second avg: 0.000
```

## 4.4 - Roll-out of Notification Server

### *4.4.1 - Uniqush install*

Basic installation method:

1. Install Redis

```
 sudo apt-get install redis-server
```

2. Install uniqush-push

- Download the latest uniqush-push binary package appropriate for your system from the download page or you can build it from source. Go 1.6+ is recommended.
- If you are using a Debian-based system, you can install the DEB:

```
sudo dpkg -i uniqush-push\_VERSION\_ARCH.deb
```

- If you are using yum/rpm-based system, you can install the RPM:

```
sudo rpm -ivh uniqush-push-VERSION-ARCH.rpm
```

- Otherwise, you can always install uniqush-push using the tarball:
    - unpack the file:

```
tar zxvf uniqush-push-VERSION-ARCH.tar.gz
```

    - enter that directory and execute *install.sh* under that directory:

```
cd uniqush-VERSION-ARCH/ && sudo ./install.sh
```

### 4.4.2 - *Application configuration directory*

The application configuration directory must be created in the folder system and set as follows:

```
/etc/uniqush/uniqush-push.conf.
```

Sections in Uniqush configuration file:

| Section | Description |
|---|---|
| **default** | Default section is the outermost section, which does not need a section name. It lays outside of all sections and some global settings exist there. |
| **WebFrontend** | Controls the web service interface of uniqush-push. Because a user of uniqush-push communicates with it through HTTP Post request, it has to specify some basic information, such as the port number, logging level, etc., to work with uniqush |
| **AddPushServiceProvider** | Contains the options used when adding a push service provider request is being processed |
| **RemovePushServiceProvider** | Contains the options used when removing a push service provider request is being processed |
| **Subscribe** | Contains the options used when a subscribe request is being processed |
| **Unsubscribe** | Contains the options used when when an unsubscribe request is being processed |
| **Push** | Contains the options used when a push request is being processed |
| **Database** | Contains the options specifying the database parameters |

Options for configuration file with default value to use:

| Option | Values | **Default** | Description |
|---|---|---|---|
| **logfile** | A file name of the log file. | **standard error** | The value of this option should be path to the log file. uniqush-push should have write permission to that file. If the file does not exist, uniqush-push} will try to create it with permission 0600. If all fails, it will write to standard error. |
| **log** | on/off | **on** | Turn on/off the logging facility of a specific part. If the log is turned off, then the logging-related options will be ignored. |
| **loglevel** | standard, verbose | **standard** | The verbosity of the log |
| **addr** | [ip address]:[port number] | **localhost: 9898** | Accepted host address and the port number of the web service. If you want to accept all connections (you probably do not want to do that), you can specify the ip address as 0.0.0.0. We recommend uniqush-push accept connections only from localhost or only from a known server. |

Options for *Database* Section

| Option | Values | Default | Description |
|---|---|---|---|
| **engine** | redis | redis | The database engine name. We support only the redis database at this point of time. So only *redis* is the valid value for this option |
| **name** | 0 | 0 | The database name. |
| **host** | *IP Address* | localhost | The host's name on which the database is running |
| **port** | *Port Number* | 0 | The port number to connect to the database program. 0 stands for the default port of the specified database engine |
| **password** | *Password* | empty string | The password of the database |
| **everysec** | *Number of seconds* | 600 | The synchronization period of cache. |
| **leastdirty** | *Number of elements* | 10 | The least number of dirty elements in cache. Combining with *everysec*option, these options tell the cache to synchronize to database every *everysec* if there are at lease *leastdirty* changes. |
| **cachesize** | *Integer Value* | 1024 | The number of elements that a cache will hold at most. |

Logfile location:

```
/var/log/uniqush/
```

### 4.4.3 - Testing

Send test message after configured.

## 4.5 - Roll-out of Mobile Applications

### 4.5.1 - Application Installation

Per Technical Design Manual section 6.3:

For Android apps:

Apps for users can be submitted to the Google play store: Play Store and deployed. Applications for use in-house are similarly deployed using the Google play store, but distributed via an enterprise account Play Store for Enterprise.

For iOS Apps:

User Apps can be submitted to the iOS App Store: App Store and downloaded by users via the Apple App Store. For in house applications (that is, applications used by shop managers and others working at OC Pizza), the applications are deployed using an Apple Business Manager account Apple Business Manager.

# 5 - Start/Shutdown procedure

## 5.1 - Webserver

In terminal type:

```
To startup server
1. cd to installation directory
2. /startup.sh

To quit (kill) server:
1. cd to installation directory
2. /shutdown.sh

To reload the server:
It is not advisable, server needs to be quit than startup again
```

## 5.2 - Database

Startup procedure:

If MySQL installation includes mysqld_safe:

```
bin/mysqld_safe --user=mysql &
```

If MySQL installation includes systemd support:

```
systemctl start mysqld
```

Shutdown procedure:

```
mysqladmin -u root -p shutdown
```

# 6 - Update procedure

## 6.1 - Webserver

Steps will be using the [old version] version to be upgraded to [new version]

1. Download latest tar.gz file
2. Uncompress the tar.gz file by running the command:

```
gunzip apache-tomcat-[new version].tar.gz.
```

3. Untar it to the /[apache dir] directory.

```
cd /[apache dir]

tar -xvf <path-to-tar-file>/apache-tomcat-[new version].tar
```

This will create the /[apache dir]/apache-tomcat-[new version] directory structure.

4. Stop the current Tomcat process by executing:

```
/[apache dir]/Tomcat_[old version]/bin/shutdown.sh
```

as the id that is currently running the Tomcat process, normally root.

5. Copy the old configuration file to the new installation:

```
cp /[apache dir]/apache-tomcat-[old version]/conf/Catalina /[apache dir]/apache-
tomcat-[new version]/conf/
```

6. Copy the old server.xml and web.xml files to the new installation:

```
cp /[apache dir]/apache-tomcat-[old version]/conf/server.xml /[apache dir]/apache-
tomcat-[new version]/conf/

cp /[apache dir]/apache-tomcat-[old version]/conf/web.xml /[apache dir]/apache-tomcat-
[new version]/conf/
```

*Note: If SSL is implemented, review the port 8443 or 443 configuration on server.xml
If there is a keystore specified, copy the file to the /[apache dir]/apache-tomcat-[new version] directory structure to match the location in the old implementation.

7. Modify the setenv.sh script located in CATALINA_HOME/bin by adding the following java options:

```
JAVA_HOME=/usr/java/default ; export JAVA_HOME <-- adapted as needed
JAVA_OPTS="-Xms4096M  -Xmx4096M  -XX:+UseConcMarkSweepGC   -XX:MaxMetaspaceSize=256M
-XX:+HeapDumpOnOutOfMemoryError  -XX:ErrorFile=/[apache dir]/apache-tomcat_[new
version]/logs/java_hs_err.log  -XX:HeapDumpPath=/[apache dir]/apache-tomcat_[new
version]/logs/  -Djava.awt.headless=true -Dsun.java2d.fontpath=<JREPATH>/lib/
```

Add the following options if you are using Oracle Java 1.8

```
XX:+UseParNewGC
XX:-UseCompressedOops
```

Check the initial memory (-Xms) and maximum memory (-Xmx) values from the old version of tomcat and use them in the new one

8. Start the new tomcat using the command:

```
/[apache dir]/apache-tomcat-[new version]/bin/startup.sh
```

## 6.2 - Database

First shut down the database:

```
mysqladmin -u root -p shutdown
```

Second, download and unpack updated distribution packages, then install

Restart MySQL:

```
mysqld_safe --user=mysql --datadir=/path/to/existing-datadir &
```

-or-

```
systemctl start mysqld
```

If there are encrypted InnoDB tablespaces, use the

```
--early-plugin-load
```

option to load the keyring plugin.

It is also possible to perform a logical upgrade, in which the complete database is exported, MySQL shut down, upgrade the installation as above, then run mysql and initialize using the new data directory. Start MySQL, update the username and password, and re-load the previous databases.

# 7 - Supervision/Monitoring

## 7.1 - Webserver

To verify that the new Apache version is running, simply start a web browser and connect to the server with http://*nodename*:8080. If you see the default Tomcat homepage everything is working correctly.

## 7.2 - Database

Use mysqladmin to verify the server is running. The following commands perform simple tests to check if the server is up and responding to connections:

```
bin/mysqladmin version
bin/mysqladmin variables
```

If you cannot connect to MySQL:

```
shell> bin/mysqladmin -u root -p version
Enter password: (enter root password here)
```

Verify you can shut down the server

-or-

```
bin/mysqladmin -u root shutdown
```

Use mysqlshow to see the existing databases:

```
bin/mysqlshow
```

To list tables within a database table:.

```
bin/mysqlshow databaseNameToShow
```

Use mysql to select information from a table in the mysql schema:

```
bin/mysql -e "SELECT User, Host, plugin FROM mysql.user" mysql.
```

This will present information from the mysql schema. If presented properly, schema is functioning properly.

# 8 - Backup and restore procedure

Automated rsync script to fully backup all servers to a remote (but online) backup destination.