|  |
| --- |
| **OC Pizza Company**  **Information Management System**  Program Roll Out Procedure  Version 1.0 |
| **Author**  Scott Bolin  *Engineer* |

Table of contents

1 -Versions 4

2 -Introduction 5

2.1 -Document purpose 5

2.2 -References 5

3 -Pre-requisites 6

3.1 -System 6

3.1.1 -Database Server 6

3.1.1.1 -Technical specifications 6

3.1.2 -Web server 6

3.1.2.1 -Technical specifications 6

3.1.3 -Batch Server 6

3.1.4 -File Server 6

3.2 -Databases 6

3.3 -Web services 6

3.4 -Other Resources 6

4 -Rollout procedure 7

4.1 -Batch Deployment 7

4.1.1 -Artifacts 7

4.1.2 -Environment variables 7

4.1.3 -Configuration 7

4.1.3.1 - Xxx.yyy file 8

4.1.3.2 - Zzz.ttt file 8

4.1.3.3 -File ... 8

4.1.4 -Resources 8

4.1.5 -Verifications 8

4.2 -Web Application rollout 9

4.2.1 -Artifacts 9

4.2.2 -Web application environment 9

4.2.2.1 -Environment variables 9

4.2.3 -Application configuration repository 9

4.2.3.1 - Xxx.yyy file 9

4.2.4 -Datasources 9

4.2.5 -Resources 10

4.2.6 -Verifications 10

5 -Start up/shutdown procedures 11

5.1 -Database 11

5.2 -Batches 11

5.3 -Web application 11

6 -Update procedure 12

6.1 -Database 12

6.2 -Batches 12

6.3 -Web application 12

7 -Supervision/Monitoring 13

7.1 -Supervision of the web application 13

8 -Back-up and restore procedure 14

9 -Glossary 15

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Description | Version |
| SB | 15/09/2020 | Creation of document | 1.00 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## 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

## References

For further information, please refer to:

1. **TF - 1**: Technical file for the application

# Prerequisites

## System

### Web Server

NGINX HTTP Server

#### 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.

There are no specific hardware requirements, thus any machine that can run the above OS/Platform is sufficient to run NGINX

### 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.

#### 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.

### Database Server

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

#### 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. A mobile app running on Android phone, iPhone, or iPad is offered with full functionality. Alternatively, and Android phone can be used with the web interface.

### Authorization Framework

OAuth 2.0 framework is used to handle authentication.

#### Technical specifications

The OAuth 2.0 framework is to be included as an external framework in the application requiring authorization capabilities. For example, in the iOS app, the Swift language framework OAuthSwift would be compiled with the Mobile application.

## Databases

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

* **MySQL** version 8

## Web services

The following web services must accessible et up to date:

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

## Other Resources

No other resources are used/needed.

# Roll-out procedure

## Roll-out of Web Server

### NGINX install

Basic install instructions:

* Install pkgsrc;
* Install and configure nginx;
* Install and configure php;
* Create a script, to start , stop , and restart , nginx and php;
* Create a launch daemon for nginx and php;
* Test everything

Typically, all this is done via package installer, with various installation scripts available on the NGINX website here: [NGINX Packages](https://nginx.org/en/linux_packages.html#RHEL-CentOS)

Full installation instructions can be found here: [Administration Guide](https://docs.nginx.com/nginx/admin-guide/).

### Environment variable

These are the environment variables recognized by the XXX application batches

|  |  |  |
| --- | --- | --- |
| Name | Obligatory | Description |
| XXX\_HOME | no | Root directory for the installation of the application |
|  |  |  |

### Configuration

These are the various configuration folders:

Server:

server {

location / {

root /data/www;

}

location /images/ {

root /data;

}

}

Proxy Server:

server {

location / {

proxy\_pass http://localhost:8080;

}

location \.(gif|jpg|png)$ {

root /data/images;

}

listen 8080;

root /data/up1;

location / {

}

}

### Verifications

In order to check that the batches are running correctly, follow these steps:

During build:

./configure --with-debug ...

Enable error log:

error\_log /path/to/log debug;

Verify nginx is configured to support debugging:

Nginx -v –with-debug

Review log for errors.

## Roll-out of Database

### MySQL Install

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

### Application configuration directory

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

C:\Program Files/MySQL

Templates:

C:\ProgramData /MySQL/Templates/

packagerules.xml:

C:\ProgramData /MySQL/

Products.xml:

C:\ProgramData /MySQL/products/

Cache:

C:\ProgramData /MySQL/Cache/

### Datasources

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

Typically the data source is located at:

/usr/local/mysql

### 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 slightly 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

## Roll-out of Notification Server

### Uniqush install

Basic installation method:

1. Install Redis

sudo apt-get install redis-server

2. Install uniquash-push

* Download the latest uniqush-push binary package appropriate for your system from the [download page](http://uniqush.org/downloads.html) 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

### 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 Name | Possible Values | Default Value | 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 Name** | **Possible Values** | **Default Value** | **Description** |
| engine | redis | redis | The database engine name. We support only the [redis](https://redis.io) 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 everysecoption, 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/

### Testing

Send test message after configured.

## Roll-out of OAuth 2 Framework

### OAuth 2 install

**Background**:

OAuth is used to authorize a third-party client (in our case, the Mobile application running on Android or iOS or on the web site, ie, the client tier) for use on a resource server (the server tier) via an authorization server. The authorization server will be the resource server in our setup. The user is the resource owner granting access to some portion of their account. In other words, the user is allowing the client tier to access their user information (or company information) on the resource server.

**Installation**:

OAuth can be set up such that the (local) server tier communicates with an external authorization server, whereby the local server passes on the authorization request to the authorization server, and the authorization server than presents to the user a login. Once the user logs in successfully, the authorization server than passes back the state to the local server and access is given.

// Talk to Peter about this, I can’t find clear instructions how to install, or maybe this ins’t even needed – authorization is handled locally only.//

# Start/Shutdown procedure

## Webserver

In terminal type:

nginx -s <quit> - shut down gracefully

nginx -s <reload> - reload the configuration file

nginx -s <reopen> - reopen log files

nginx -s <stop> - shut down immediately (fast shutdown)

nginx -s <kill> - kill the master process immediately

## 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

## Other

# Update procedure

## Webserver

For Amazon Linux, CentOS, Oracle Linux, and RHEL:

sudo yum upgrade nginx

For Debian and Ubuntu:

sudo apt-get update

sudo apt-get install nginx

To verify that the new NGINX version is running:

Nginx-v

Nginx version: nginx/1.19.0 (nginx-plus-r22) [for example]

## 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](https://dev.mysql.com/doc/refman/8.0/en/server-options.html#option_mysqld_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.

## Other

# Supervision/Monitoring

## Webserver

To verify that the new NGINX version is running:

Nginx-v

Nginx version: nginx/1.19.0 (nginx-plus-r22) [for example]

## 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.

## Other

# Backup and restore procedure

# Glossary

|  |  |
| --- | --- |
|  |  |
|  |  |