

# How to use Heroku over Maven for Deployment your Application

## What you need ?

- You need API KEY : from HEROKU
- You need to Download <https://github.com/heroku/heroku-maven-plugin>
- You need HEROKU CLI : "The Heroku CLI"

Add heroku to your maven file :

```
[crayon-66351a99bb454633669660/]
```

```
[crayon-66351a99bb459604316900/]
```

# If you are not using git, To define the app name you should use this configuration

```
[crayon-66351a99bb45c448428933/]
```

**\* This is enough for WAR package.**

**# BUT -> If the Application is stand alone, you should set how the application should be run:**

```
[crayon-66351a99bb45e072896357/]
```

Deploy the WAR

```
[crayon-66351a99bb460121582336/]
```

Check it is deployed and running

```
[crayon-66351a99bb462349675537/]
```

Extra Configuration

```
[crayon-66351a99bb464048978324/]
```

—

— ffa ds

—

ref: <https://github.com/heroku/heroku-maven-plugin>

---

# HTML / CSS

## CSS Notes

[How to Horizontally Center Div Content \(Flex, Display, Float\)](#)

\*

\*

\*

---

# Postgresql

[Sql Queries From 0 to Professional](#)

[UUID type and Auto Generation](#)

[Disk Size Restore](#)

[Schema Creation / Table](#)

[User Creation and Role Assigns](#)

- - - - -

Centos 7 – Installation Postgres 12

[Centos 7 – Installation Postgres 14](#)

- - - - -

- - - - -

- - - - -

---

## Oracle Data Catalog – 1

## Current Capabilities

### Metadata Harvesting

- Searchable data asset Inventory
- OCI Object Storage, Autonomous Database
- Oracle DB, MySQL, Hive, Kafka on OCI & on-prem

### Metadata Curation

- Business Glossaries with Terms and Categories
- Tags for annotations
- Link assets to business terms, tags

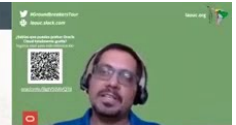
### Search and Browse

- Collaborative environment
- Search based on technical names, business terms, tags
- Data assets hierarchy

### Optimized for Oracle Cloud

- Secure, scalable, serverless cloud-native
- REST APIs and SDKs in Java, Python, Ruby, Go
- IAM based policy management

**\* Current capabilities available at no cost**



# Linux Package Installation & Deployments

\* Rust Deployment & Installation

\* Rust Repositories

\* [Git Practices](#)

\* [Jenkins Create Maven Auto Build-1](#)

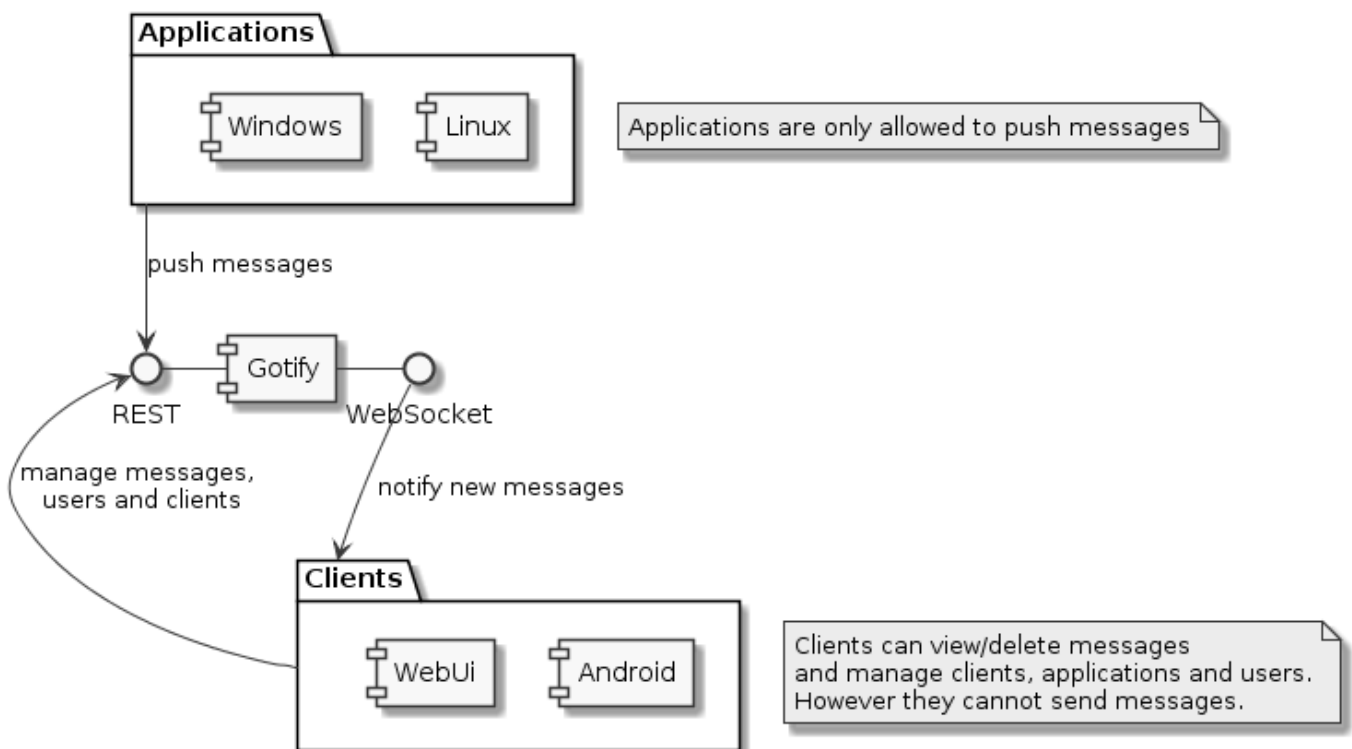
\* [Heroku \(Deployment from Git – 1 \)](#)

\* [How to use Heroku over Maven for Deployment your Application](#)

---

# Push Notification Server Installation + Android APP + Channel

Gotify : Litesql using, but if you want using : Nginx



Gotify Proxy Config : Arka planda olursa server  
[crayon-66351a99bbed0795433285/]

## Gotify Android APP

[github.com/gotify/android](https://github.com/gotify/android)

## Server

<https://github.com/gotify/server>

## Swagger

gotify.net/api-docs#/message/createMessage

PHP posting eXample

[crayon-66351a99bbed4553312825/]

.

---

# Kubernetes CNI

## What is CNI?

**CNI** is short for **Container Networking Interface**, which is basically an external software (module) that implements an [interface well defined by a specification](#), which allows **Kubernetes** to perform actions to provide network functionality.

*“Each CNI plugin must be implemented as an executable that is invoked by the container management system (e.g. rkt or Kubernetes). A CNI plugin is responsible for inserting a network interface into the container network namespace (e.g. one end of a veth pair) and making any necessary changes on the host (e.g. attaching the other end of the veth into a bridge). It should then assign the IP to the interface and setup the routes consistent with the IP Address Management section by invoking appropriate IPAM plugin.”*

*Reference:*

<https://github.com/containernetworking/cni/blob/master/SPEC.md#cni-plugin>