



# Web Deployment

## Web Engineering

Prof. D. König



# 3 Deployment Options



upload war file to Java EE web server,  
e.g. Tomcat via manager-gui

upload war file to a PaaS provider via CLI

upload executable to an IaaS provider



# Preparation 1



adapt grails-app/conf/application.yml

production:

dataSource:

dbCreate: ~~none~~ create

*only on first deployment!*



# Deployment Option 1

grails war


-> *build/libs/appname-version.war*


upload war file to Java web server,  
e.g. Tomcat



localhost:8080/manager/html

DuckDuckGo    Yahoo!    Google Maps    YouTube    Wikipedia    News    Beliebt    Chatter    Hoogle    Category: Th...amming Cafe    Hulbee    Froogle    Home | grails...munity Slack





# Tomcat Web Application Manager

Message:OK

Manager

List Applications

HTML Manager Help

Manager Help

Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessionswith idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessionswith idle ≥ 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessionswith idle ≥ 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessionswith idle ≥ 30 minutes</div>
/manager	None specified	Tomcat Manager Application	true	1	<div>StartStopReloadUndeploy</div> <div>Expire sessionswith idle ≥ 30 minutes</div>

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload

Datei auswählen

Keine Datei ausgewählt

Deploy

# Deployment Option 2

`grails war`

upload war file to a Platform-as-a-Service (PaaS) provider, e.g. Heroku, AWS, ...

via command line interface (CLI) or web

# Heroku



Sign up for free under [heroku.com](https://heroku.com)

Follow the CLI installation procedure

Create a new app through the web interface



# Deploy from CLI

heroku login

heroku plugins:install heroku-cli-deploy

heroku war:deploy *file* --app *app-name*

<https://devcenter.heroku.com/articles/war-deployment#deployment-with-the-heroku-cli>



# Deployment Option 3

gradlew stage

upload self-starting application to a  
Infrastructure-as-a-Service (IaaS) provider,  
e.g. Heroku

# Preparation

in /build.gradle

task copyDistToHerokuUp

rename build/install/WebEngineering/

to build/install/<yourProjectName>/

# gradlew stage

assembles the app distribution

copies it ../heroku-up

(this is the app to deploy)

# Adaption

in ../heroku-up

add a file Procfile (*no extension*)

in that file, add the line (all on one line)

```
web: export JAVA_OPTS=-Dserver.port=$PORT; ./
bin/<yourProjectName>/
```

# Heroku



Sign up for free under [heroku.com](https://heroku.com)

Follow the installation procedure

Note the coordinates (git repo, URL)

# Prepare Deployment

```
cd ../heroku-up
```

```
heroku create // note the coordinates
```

```
git push heroku master
```

```
heroku ps:scale web=1
```

```
heroku open
```

# Push a New Version

```
gradlew stage
```

```
cd ../heroku-up
```

```
git commit -am "my newest version ..."
```

```
git push heroku master
```



# Validating via git

git status *// have I committed all files?*

git remote -vv *// what remotes do I have?*

# Optional: Custom Domain

<https://fhnw-webec.herokuapp.com>

# Web Atrocities

- Care for your data (download, backup)
- Data harvesting
- Data protection
- Forgetful data

# Web Atrocities

- Denial of Service attacks, Spamming
- Proper Password handling, encryption
- Logging, Monitoring
- Localization  
formats, timezones, character encoding