

Deployment

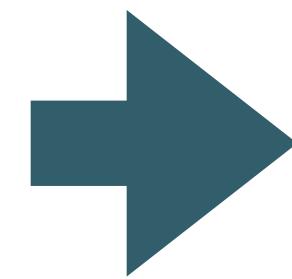
Lab-12c
Deploy



Deploy a Play Application to
the cloud

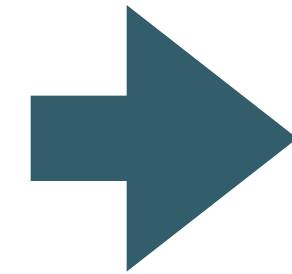
Application runs in 2 Modes

Development Mode



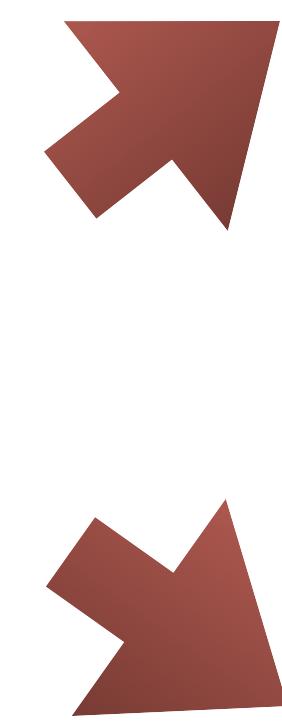
Application runs on
developer workstation.
Accessible locally

Production Mode

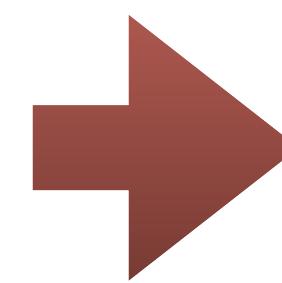


Application runs on
cloud servers.
Accessible globally

Production
Mode

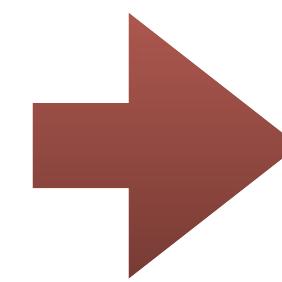


Play Application runs on
Application Server



<https://heroku.com>

Database runs on
Database Service



<https://www.elephantsql.com>

Development Mode

The screenshot illustrates the development environment for a Play Framework application named 'todolist'. The application structure is as follows:

- Project Tree:** Shows the 'todolist' project with its sub-modules: docviewer (~/dev/play-1.3.4/modules/docview), app, conf, public, test, tmp, and various configuration files like application.conf, data.yml, dependencies.yml, messages, and routes.
- Code Editor:** Displays the 'Dashboard.java' file under the 'app/controllers' package. The code implements a 'Dashboard' controller with methods for rendering the index page, adding todos, and deleting todos.
- Browser:** Shows the application running at localhost:9000/dashboard. The page title is 'Dashboard'. It displays a 'Todo List' for 'homer simpson's Todo List' with two items: 'Make tea' and 'Go for snooze', each with a 'Delete' button. Below the list is a form for adding a new todo item.
- Terminal:** Shows the command 'play run' being entered into the terminal. The output indicates the application is running on port 9000, using Java version 1.8.0_162, and listening for transport dt_socket at address 8000. It also shows a warning about the DATABASE_URL configuration.

A large blue arrow points upwards from the terminal output towards the browser window, indicating the flow of development: code changes in the IDE lead to updates in the running application.

Application running on <http://localhost:9000>

Development Mode

The screenshot shows a Java IDE interface with two main windows:

- Left Window (IDE View):** Displays the project structure for a Play Framework application named "todolist". The code editor shows the file `Dashboard.java` which contains Java code for a controller. The terminal below shows the command `play run` being entered.
- Right Window (H2 Console):** A web-based database browser showing the schema of an H2 database. It includes sections for "Important Commands" and "Sample SQL Script".

A large blue arrow points from the "Database visible on" text at the bottom right towards the H2 Console window.

Dashboard.java Content:

```
package controllers;
import ...
public class Dashboard extends Controller
{
    public static void index()
    {
        Logger.info( message: "Rendering Dashboard");
        Member member = Accounts.getLoggedInMember();
        List<Todo> todolist = member.todolist;
        render( ...args: "dashboard.html", member, todolist);
    }

    public static void addTodo(String title)
    {
        Member member = Accounts.getLoggedInMember();
        Todo todo = new Todo(title);
        member.todolist.add(todo);
        member.save();
        Logger.info( message: "Adding Todo" + title);
        redirect( url: "/dashboard");
    }

    public static void deleteTodo(Long id, Long todoid)
    {
        Member member = Member.findById(id);
        Todo todo = Todo.findById(todoid);
        member.todolist.remove(todo);
        member.save();
        todo.delete();
        Logger.info( message: "Deleting " + todo.title);
        redirect( url: "/dashboard");
    }
}
```

H2 Console - Important Commands:

?	Displays this Help Page
Ctrl+Enter	Shows the Command History
Ctrl+Space	Executes the current SQL statement
Shift+Enter	Executes the SQL statement defined by the text selection
Ctrl+Space	Auto complete
Disconnect icon	Disconnects from the database

H2 Console - Sample SQL Script:

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
with ID and NAME columns	
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...

Terminal Output:

```
play run
~
~ play! 1.5.0, https://www.playframework.com
~
~ Ctrl+C to stop
~
~ using java version "1.8.0_162"
Listening for transport dt_socket at address: 8000
May 03, 2018 9:10:59 AM play.Logger warn
WARNING: Cannot replace DATABASE_URL in configuration (db=${DATABASE_URL})
09:10:59,374 INFO ~ Starting /Users/edeleastar/repos/wit-hdip-comp-sci/web-dev-projects/todolist
09:10:59,468 WARN ~ You're running Play! in DEV mode
09:10:59,558 INFO ~ Listening for HTTP on port 9000 (Waiting a first request to start) ...
~ Server is up and running
```

Event Log:

10:14 LF⁺ UTF-8⁺ Git: master 5

Production Mode : Application

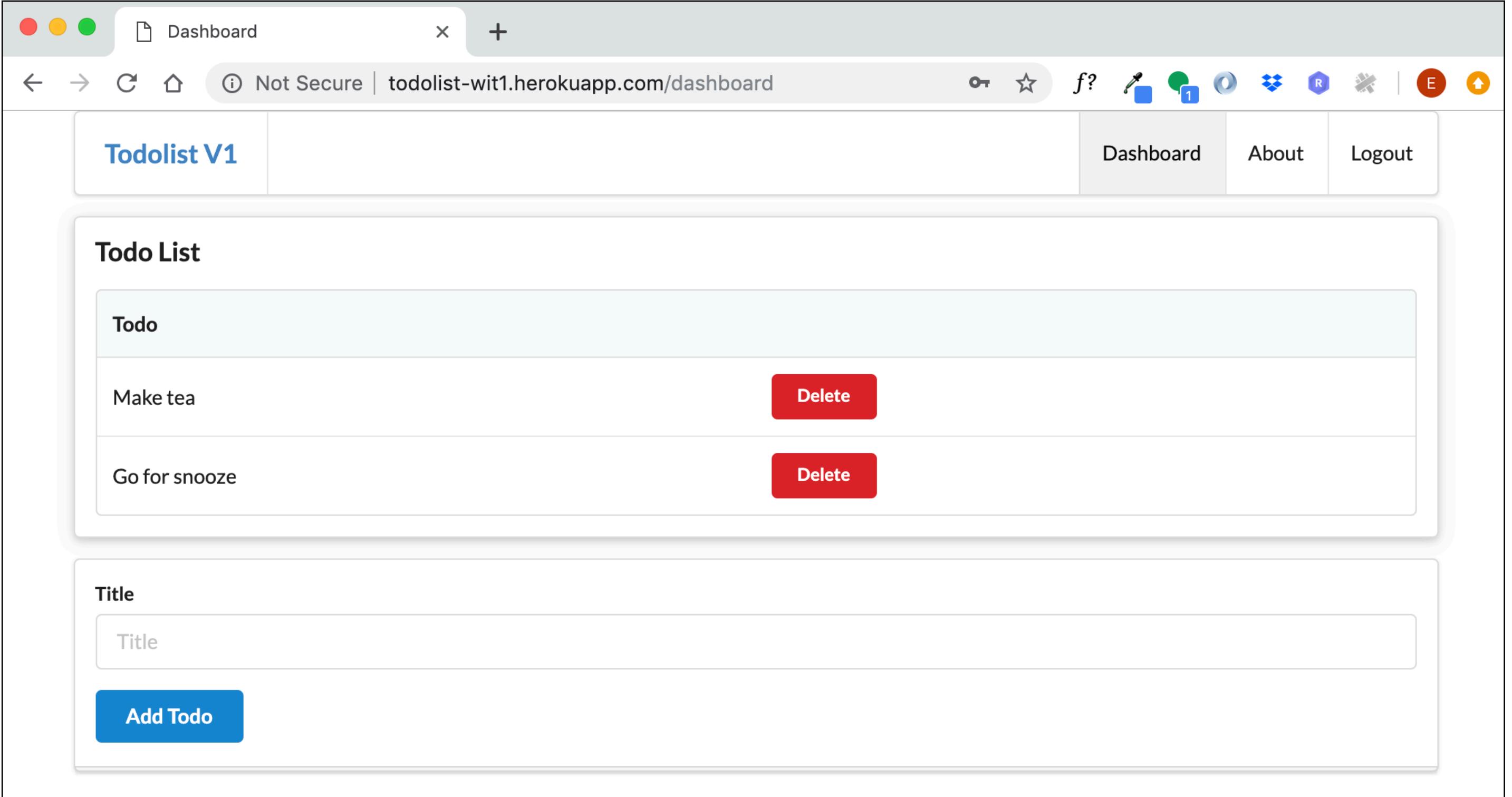
Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

 master

[Deploy Branch](#)



The screenshot shows a web browser window with the title "Dashboard" and the URL "Not Secure | todolist-wit1.herokuapp.com/dashboard". The main content area is titled "Todolist V1". It contains a "Todo List" section with two items: "Make tea" and "Go for snooze", each accompanied by a red "Delete" button. Below the list is a "Title" input field and a blue "Add Todo" button.

Application running on
<http://todolist-wit1.herokuapp.com>
(For example)

Production Mode : Database

The screenshot shows the ElephantSQL web interface. On the left, a sidebar lists various database management options: DETAILS, ALARMS, BROWSER, STATS, SLOW QUERIES, BACKUP, LOG, METRICS, ADMIN, INTEGRATIONS, and API access. The DETAILS tab is selected. In the main area, the 'Details' page is displayed, showing configuration details for a database named 'TodoList'. The database is connected to a server at 'tai.db.elephantsql.com' in the 'amazon-web-services' region, created on '2021-04-28 12:20 UTC'. It has a user 'iwbagwwl' and a password '2DWFQj...'. The URL is 'postgres://iwbagwwl:wl@...'. The current database size is 0 B and the max database size is 20 MB. Below this is an 'API access' section with an 'HTTP API' link and an 'API Key' field containing '533b1f...'. On the right, a modal window titled 'SQL Browser' is open, displaying the results of the query 'SELECT * FROM "public"."member" LIMIT 100'. The results show two rows of data:

id	email	firstname	lastname	password
4	homer@simpson.com	homer	simpson	secret
5	marge@simpson.com	marge	simpson	secret

Database console available on
<https://api.elephantsql.com/console/xxxxxxxx-yyyyy-zzzz/browser?>

https://en.wikipedia.org/wiki/Software_deployment

Software deployment is all of the activities that make a **software system** available for use.

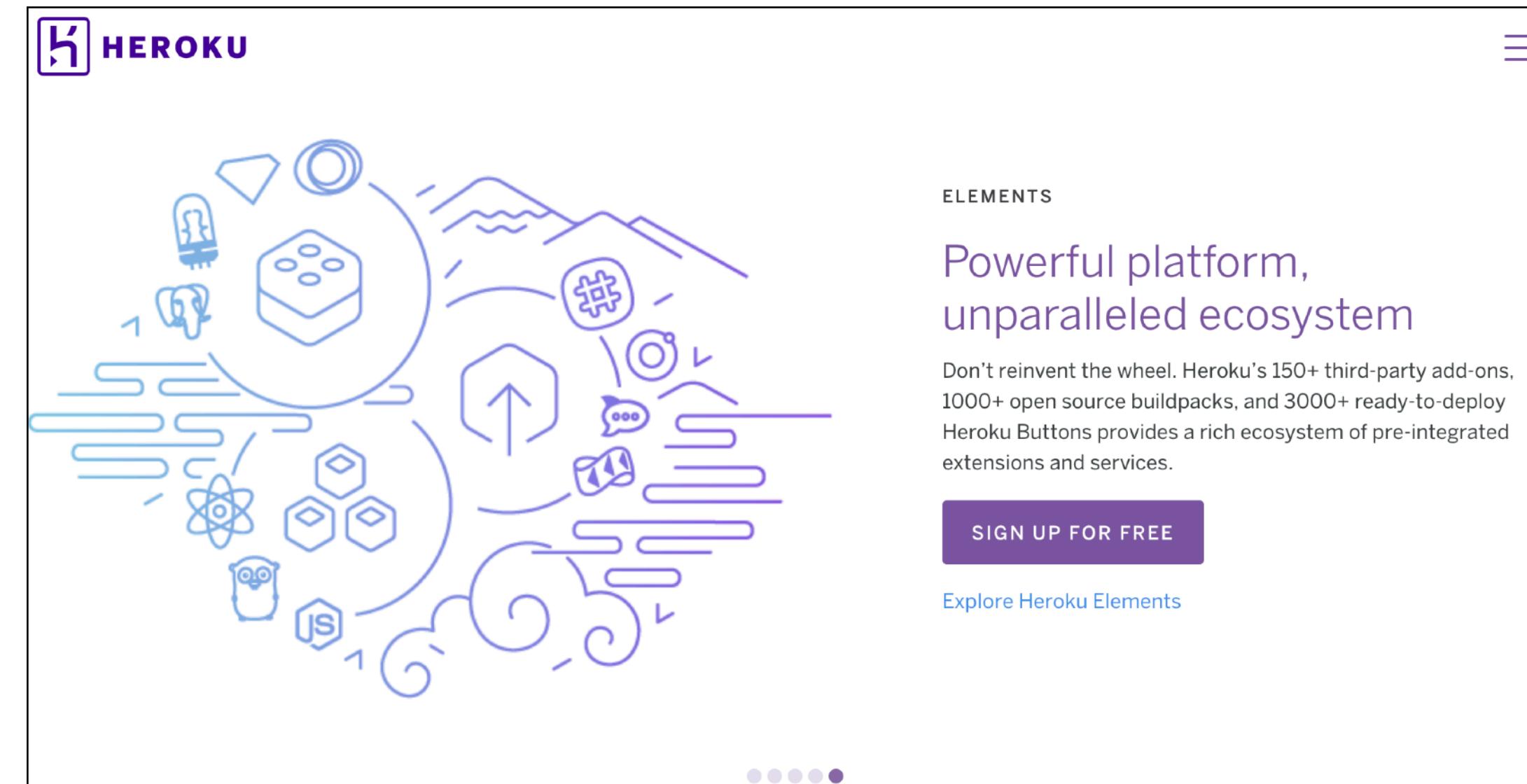
The general deployment process consists of several interrelated activities with possible transitions between them.

Transition from *Development Mode* to *Production Mode*



Deployment: Platforms & Tools

Heroku: Host for Play Application



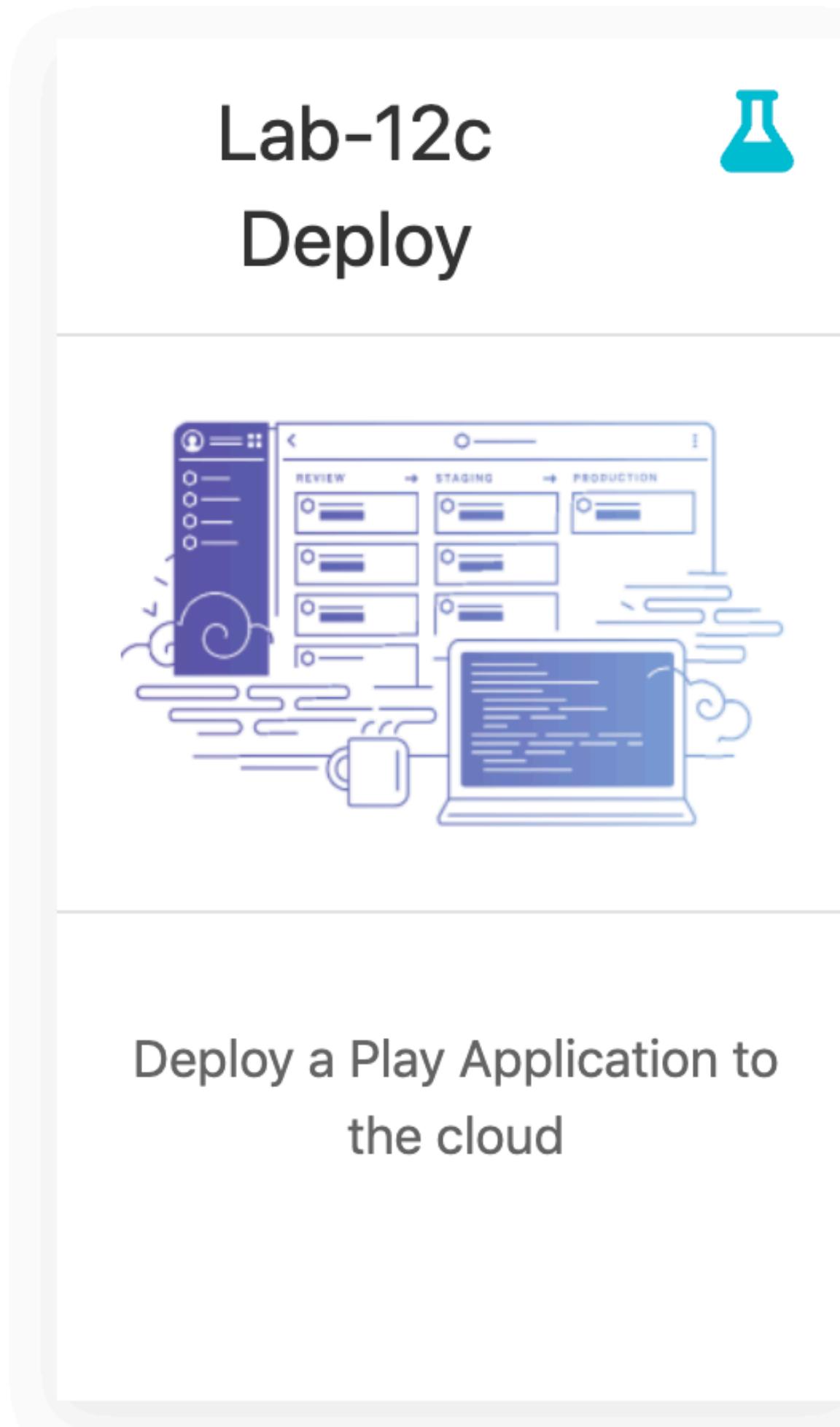
Github: Host for the application source

Elephant SQL: Host for SQL Database

The ElephantSQL interface shows detailed information for a database instance named "TodoList". The "Details" tab is active, displaying fields such as Server (tai.db.elephantsql.com), Region (amazon-web-services:eu-west-1), Created at (2021-04-28 12:20 UTC+00:00), User & Default database (iwbagwwl), Password (2DWFOJ...), URL (postgres://iwbagwwl:2DWFOJ...@tai.db.elephantsql.com:5432/iwbagwwl), Current database size (20 MB), and Max database size (20 MB). A sidebar on the right shows the "Active Plan" (Tiny Turtle) and a "Upgrade Instance" button.

The "API access" section allows controlling the instance via the HTTP API, with an API Key field containing "533b1f...".

Deploying a Play Application



1. Create Database on Elephant SQL

2. Test Local Application connected to Elephant SQL

3. Prepare Local Application for Heroku Deployment

4. Deploy Application to Heroku

Instances

Name	Plan	Datacenter
You don't have any instances yet, do you want to create one?		

MENU

- Home
- Plans
- Documentation
- Blog
- About

MORE

- Status
- Terms of Service
- Program Policies
- Privacy Policy
- Imprint

ElephantSQL

[Contact Support](#)

Open 24 hours a day, 7 days a week

© Copyright 2012-2021 ElephantSQL. PostgreSQL and the PostgreSQL Logo are trademarks of PostgreSQL Community Association of Canada.

TodoList

HDip2021

DETAILS

Details

Server	tai.db.elephantsql.com (tai-01)
Region	amazon-web-services:eu-west-1
Created at	2021-04-28 12:20 UTC+00:00
User & Default database	iwbagwwl
Password	2DWFQj...  
URL	postgres://iwbagwwl:2DWFQj...@tai.db.elephantsql.com:5432/iwbagwwl  
Current database size	
Max database size	20 MB

API access

Control your instance via our [HTTP API](#).

API Key [533b1f...](#)  

1.Create Database on Elephant SQL

Sign in to ElephantSQL with Github Account

Create a Database Instance

Copy Database Connection String

2. Test Local Application connected to Elephant SQL

Edit existing file:

app/conf/application.conf

```
db.default=mem
```

Using the database in your clipboard, paste it in instead of the `mem` entry (keeping `db=`)

```
db=postgres://iwbagwwl:2DWjtsdfgsfgsdfgI1revvB3@tai.db.elephantsql.com:5432/iwbagwwl
```

Add the following entries immediately after the above:

```
jpa.dialect=org.hibernate.dialect.PostgreSQLDialect  
jpa.ddl=update
```

Save and restart the application. Notice that in the console we should see something like this:

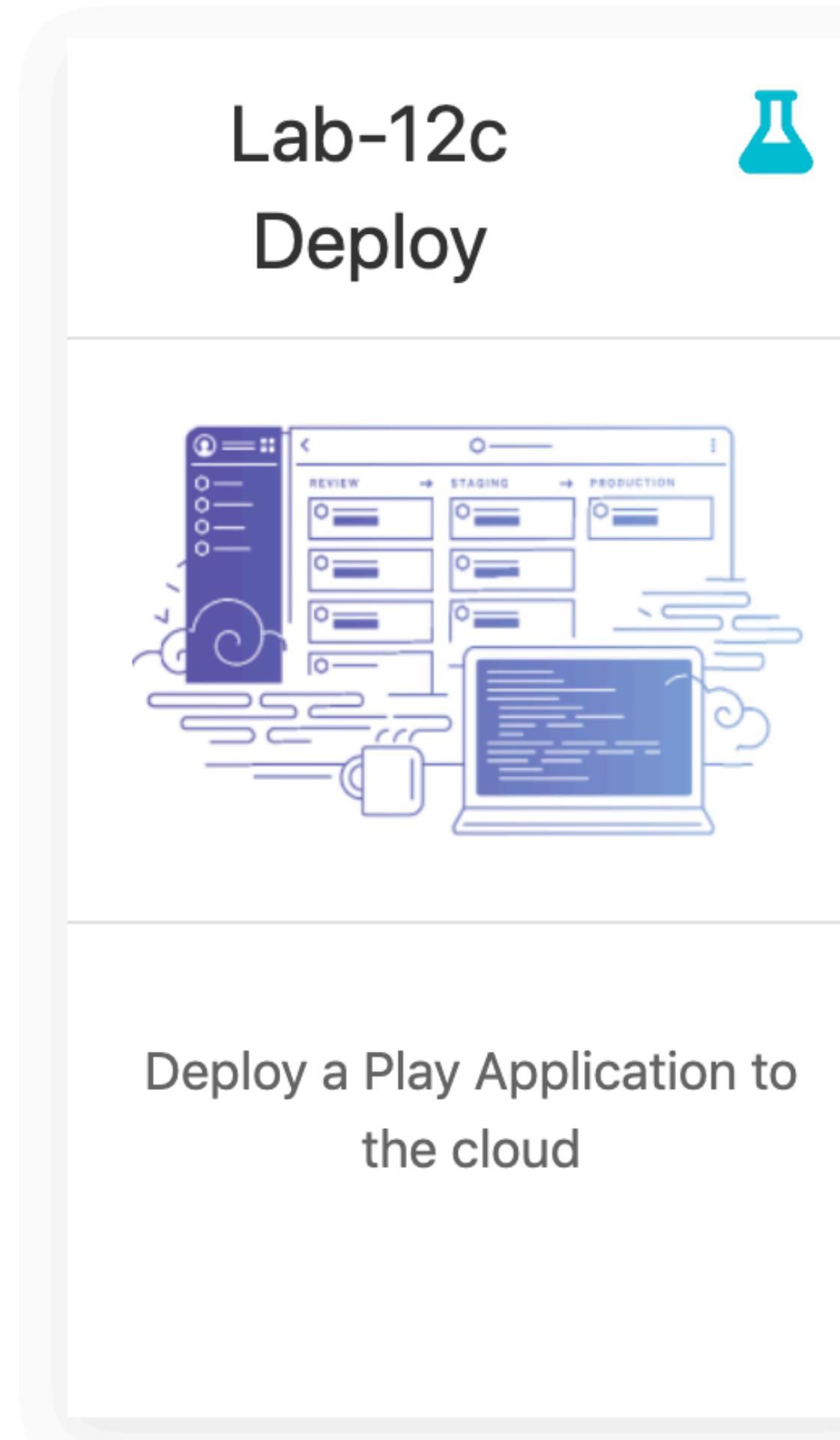
```
13:31:48,504 INFO ~ Connected to jdbc:postgresql://tai.db.elephantsql.com:5432/iwbagwwl for default
```

Dev mode
Application
connected to in
memory database

Staging: connect
local application to
Cloud Database

Run and test the
application

3. Prepare Local Application for Heroku Deployment



Three Key Configuration Parameters

- JDK Version
- Play Version
- Set Production Mode

Commit these changes:

- Push changes to Github

JDK Version

Heroku supports multiple versions of the JDK

3.Prepare Local Application for Heroku Deployment

<https://devcenter.heroku.com/articles/java-support>

Supported Java versions

Heroku currently uses OpenJDK 8 to run your application by default. OpenJDK versions 14, 13, 12, 11, and 7 are also available. Depending on the major version you select the latest available update of that JDK will be used each time you deploy your app.

Current default versions are:

- Java 7 - [1.7.0_262](#)
- Java 8 - [1.8.0_252](#)
- Java 11 - [11.0.7](#)
- Java 12 - [12.0.2](#)
- Java 13 - [13.0.3](#)
- Java 14 - [14.0.1](#)

The JDK that your app uses will be included in the slug, which will affect your slug size.

Place this file in project root:

system.properties

`java.runtime.version=11`

Play Version

3.Prepare Local Application for Heroku Deployment

<https://www.playframework.com/download#alternatives>

Play Application Framework Versions

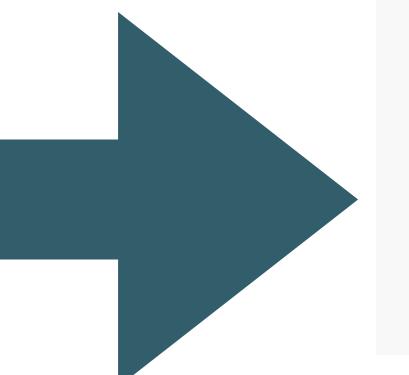
1.5 Setup Instructions

play-1.5.3.zip	Apr 8 2019	79M
play-1.5.2.zip	Oct 30 2018	79M
play-1.5.1.zip	Jul 17 2018	79M
Show all versions		

Edit existing file:

conf/dependencies.yml

```
# Application dependencies  
require:  
  - play
```



```
# Application dependencies  
require:  
  - play 1.5.3
```

Production Mode

3. Prepare Local Application for Heroku Deployment

Edit existing file:

```
# Application mode
#
# Set to dev to enable instant reloading and other development help.
# Otherwise set to prod.
application.mode=dev
%prod.application.mode=prod
```

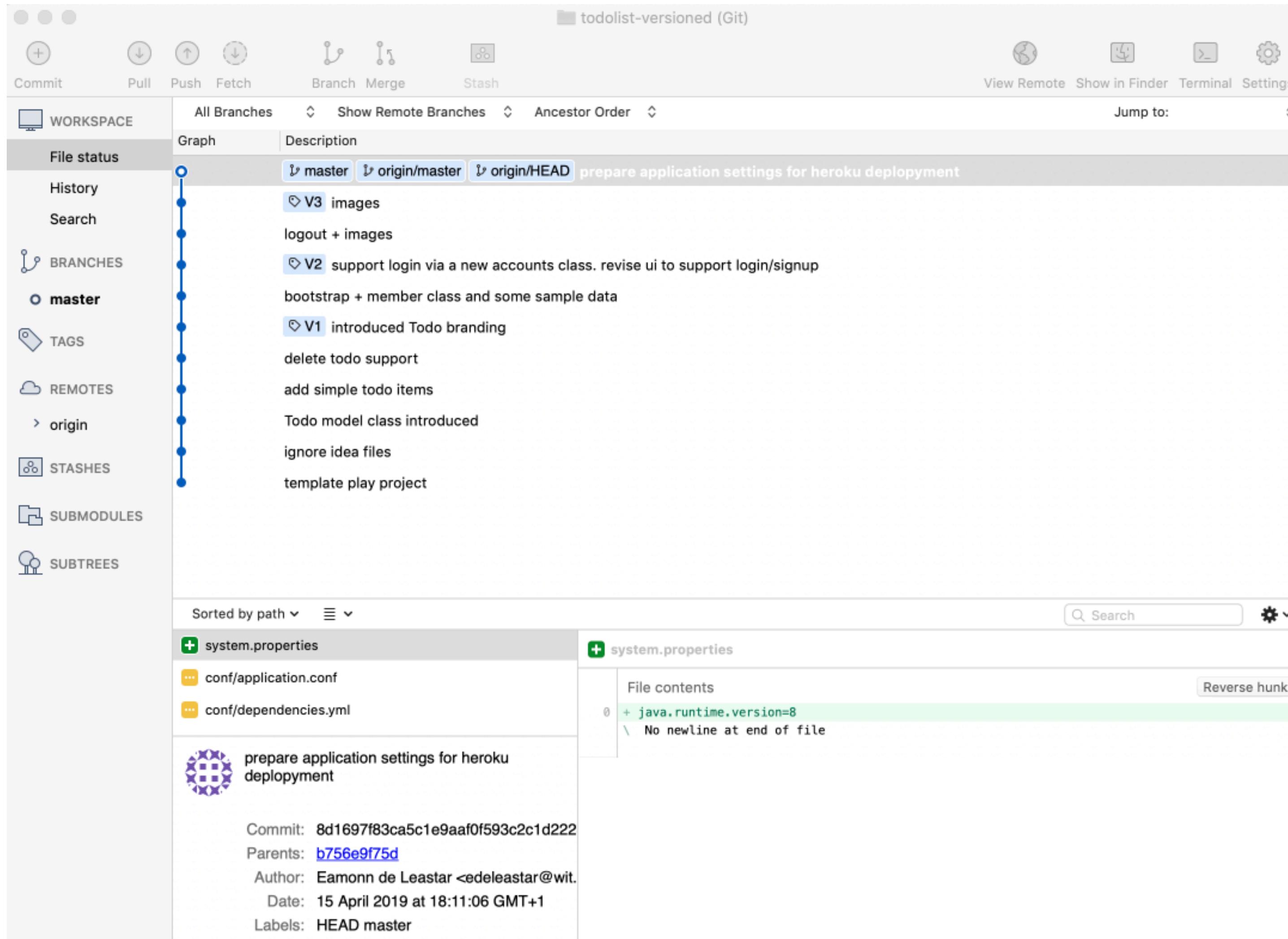
This needs to be changed to this:

```
# Application mode
#
# Set to dev to enable instant reloading and other development help.
# Otherwise set to prod.
#application.mode=dev
prod.application.mode=prod
```

Production Mode
optimises additional
features

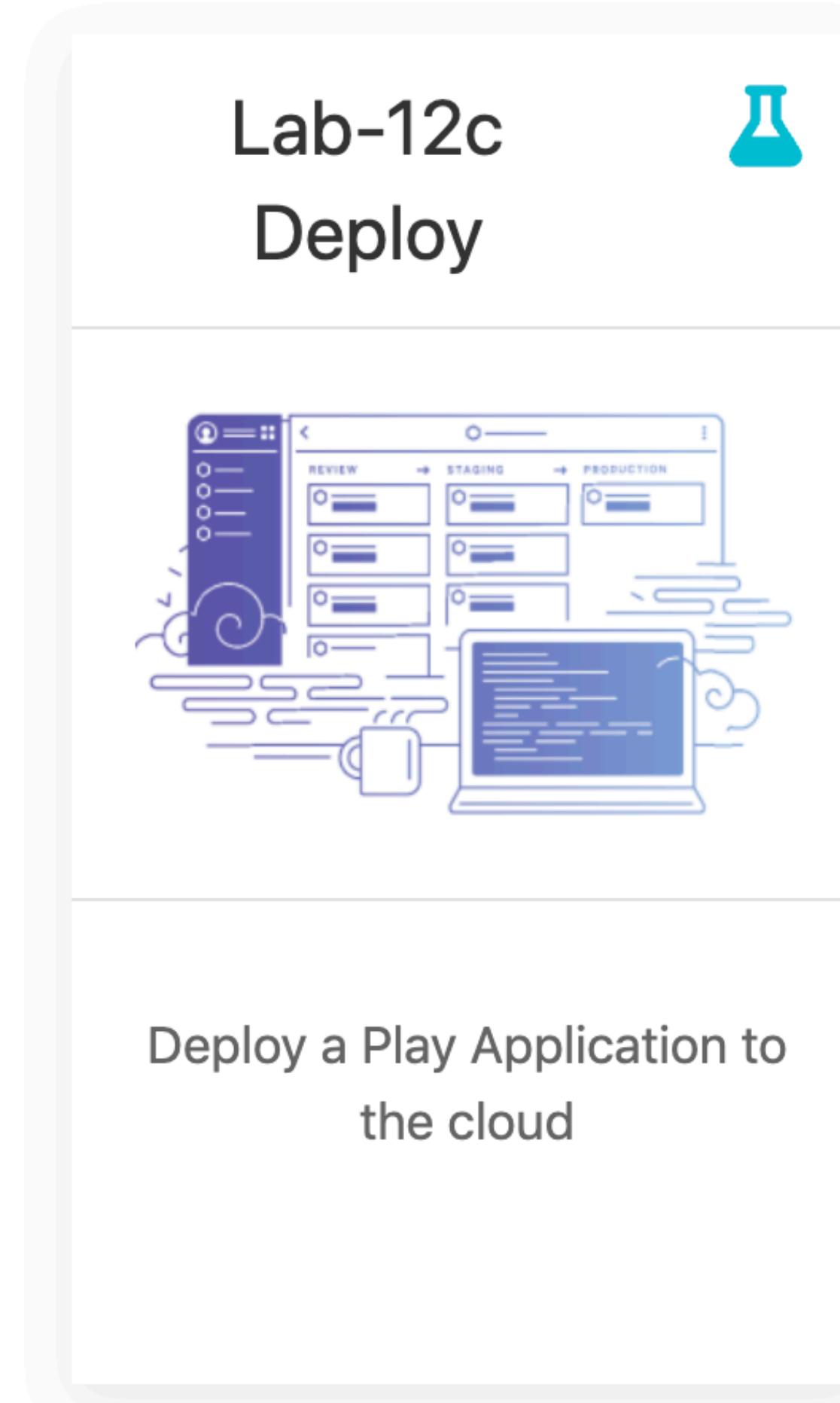
Push Changes to Github

3. Prepare Local Application for Heroku Deployment



All changes should be on GitHub version of project

4. Deploy Application to Heroku



From Command Line

- Login to Heroku
- Create the Application

From Heroku Dashboard

- Link Project to Github Repo
- Deploy application
- Monitor Logs

Login & Create Application

4. Deploy Application to Heroku

From a command shell inside your project, make sure you are logged in to Heroku:

```
heroku login
```

(You may already be logged in)

Still in the command line, enter the following:

```
heroku create --stack heroku-18 --buildpack https://github.com/heroku/heroku-buildpack-play
```

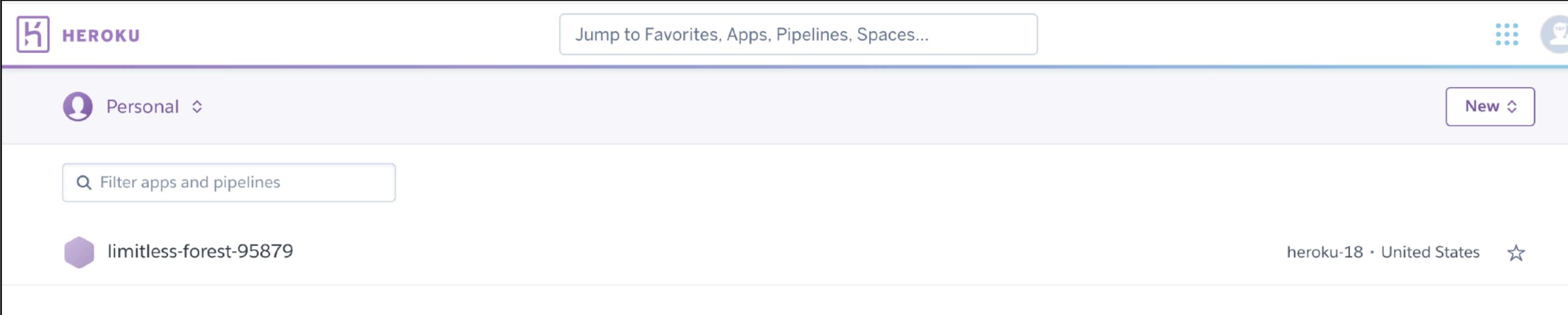
This will respond with something like:

```
Creating app... done, ⚡ limitless-forest-95879, stack is heroku-18
Setting buildpack to https://github.com/heroku/heroku-buildpack-play... done
https://limitless-forest-95879.herokuapp.com/ | https://git.heroku.com/limitless-forest-95879.git
```

Application
created on
Heroku
Cloud

Link Project to Github Repo

4. Deploy Application to Heroku



Deployment method



Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub

Connect your app to a GitHub rep

Deploy changes with GitHub

Connecting to a repository will allo

Automatic deploys from GitHub

Select a branch to deploy automa

Create review apps in pipelines

Pipelines connected to GitHub can

[Connect to GitHub](#)

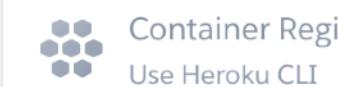
Deployment method



Heroku Git
Use Heroku CLI



GitHub
Connect to GitHub



Container Registry
Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code
diffs and deploys.

Search for a repository to connect to

wit-hdip-comp-sci-2021

repo-name

Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access](#).

wit-hdip-comp-sci-2021/todo-list-versioned

Connect

wit-hdip-comp-sci-2021/todo-list

Connect

wit-hdip-comp-sci-2021/weather-top

Connect

wit-hdip-comp-sci-2021/wit-hdip-comp-sci-2021.github.io

Connect

wit-hdip-comp-sci-2021/todo

Connect

wit-hdip-comp-sci-2021/weather-top-experiment

Connect

Deploy Application

4. Deploy Application to Heroku

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

Choose a branch to deploy

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

Choose a branch to deploy

Receive code from GitHub

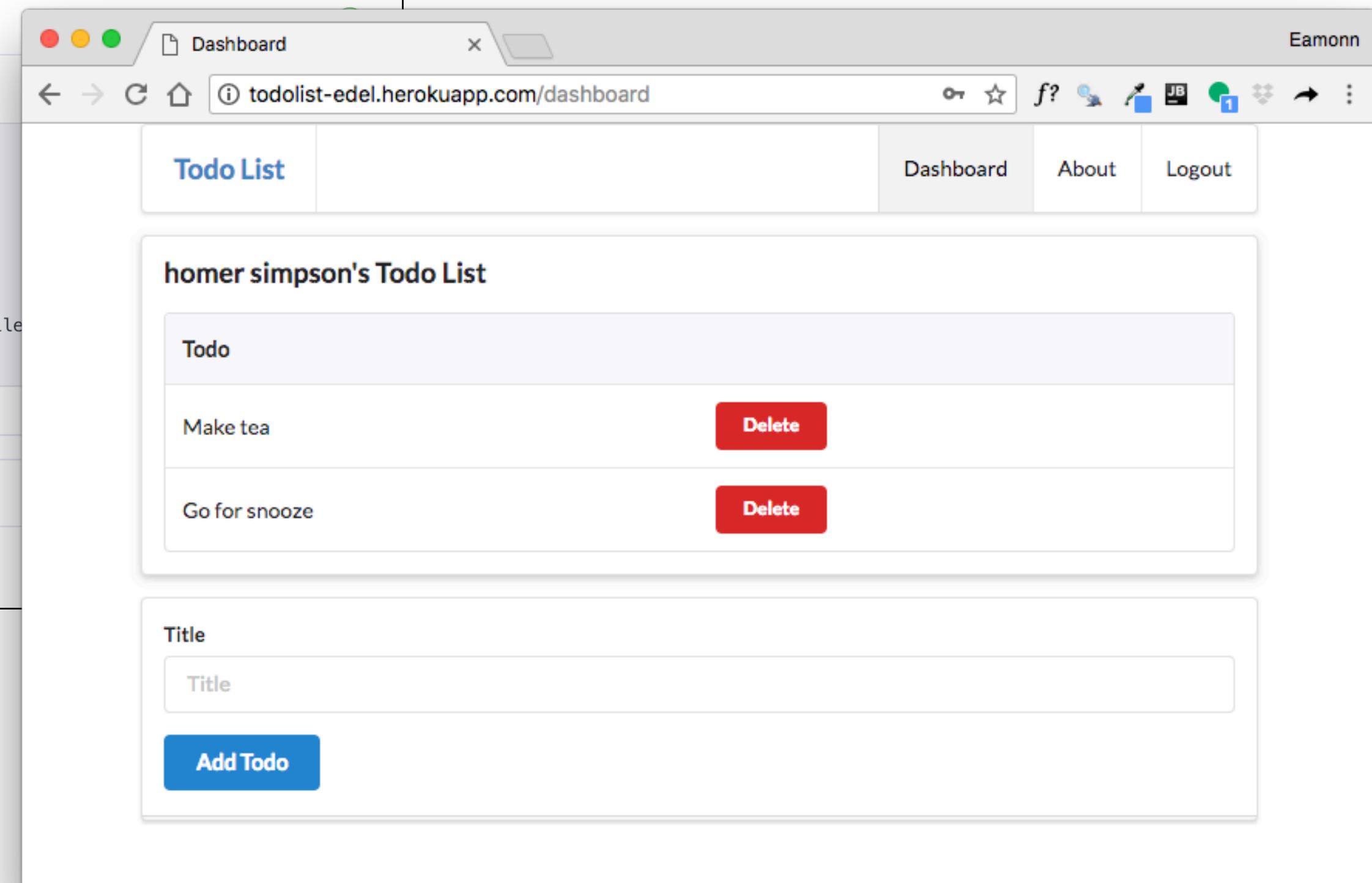
Build master f9d91660

```
~ |__| |__/
~ 
~ play! 1.5.3, https://www.playframework.com
~
1.5.3
Building Play! application at directory ./
Resolving dependencies: ./play/play dependencies ./ --forProd --forceCopy --sile
Duser.home=/tmp/build_ecd2b2a1 2>&1
```

Autoscroll with output

Release phase

Deploy to Heroku



Monitor Logs

4. Deploy Application to Heroku

Open app More ▾

- View logs
- View webhooks
- Run console
- Production check
- Restart all dynos
- Add to pipeline...

Application Logs ALL PROCESSES ▾

```
2018-04-28T06:46:57.262005+00:00 heroku[web.1]: Starting process with command `play run --http.port=8575 --%prod -Dprecompiled=true`
2018-04-28T06:46:59.289767+00:00 app[web.1]: Create a Procfile to customize the command used to run this process: https://devcenter.heroku.com/articles/procfile
2018-04-28T06:46:59.580802+00:00 app[web.1]: Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
2018-04-28T06:47:00.525471+00:00 app[web.1]: 06:47:00,524 INFO ~ Starting /app
2018-04-28T06:47:00.960385+00:00 app[web.1]: :: loading settings :: url = jar:file:/app/.play/framework/lib/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
2018-04-28T06:47:01.491954+00:00 app[web.1]: 06:47:01,491 INFO ~ Application is precompiled
2018-04-28T06:47:04.238824+00:00 app[web.1]: 06:47:04,237 INFO ~ HikariPool-1 - Starting...
2018-04-28T06:47:04.457666+00:00 app[web.1]: 06:47:04,457 INFO ~ HikariPool-1 - Start completed.
2018-04-28T06:47:04.477044+00:00 app[web.1]: 06:47:04,476 INFO ~ Connected to jdbc:postgresql://ec2-79-125-14-195.eu-west-1.compute.amazonaws.com:5432/dcbjsg53rodgeo for default
2018-04-28T06:47:07.963782+00:00 app[web.1]: 06:47:07,963 INFO ~ Application 'todolist' is now started !
2018-04-28T06:47:08.830163+00:00 app[web.1]: 06:47:08,825 WARN ~ Precompiled template /conf/data.yml not found, trying to load it dynamically...
2018-04-28T06:47:08.830174+00:00 app[web.1]: java.lang.RuntimeException: Cannot load precompiled template /conf/data.yml
2018-04-28T06:47:08.830176+00:00 app[web.1]: at play.templates.BaseTemplate.loadPrecompiled(BaseTemplate.java:44)
2018-04-28T06:47:08.830177+00:00 app[web.1]: at play.templates.TemplateLoader.load(TemplateLoader.java:75)
2018-04-28T06:47:08.830178+00:00 app[web.1]: at play.test.Fixtures.loadModels(Fixtures.java:223)
2018-04-28T06:47:08.830180+00:00 app[web.1]: at play.test.Fixtures.loadModels(Fixtures.java:191)
2018-04-28T06:47:08.830181+00:00 app[web.1]: at Bootstrap.doJob(Bootstrap.java:16)
2018-04-28T06:47:08.830183+00:00 app[web.1]: at play.jobs.Job.doJobWithResult(Job.java:64)
2018-04-28T06:47:08.830184+00:00 app[web.1]: at play.jobs.Job$2.apply(Job.java:224)
2018-04-28T06:47:08.830185+00:00 app[web.1]: at play.db.jpa.JPA.withTransaction(JPA.java:285)
2018-04-28T06:47:08.830187+00:00 app[web.1]: at play.db.jpa.JPA.withinFilter(JPA.java:238)
2018-04-28T06:47:08.830188+00:00 app[web.1]: at play.db.jpa.JPAPlugin$TransactionalFilter.withinFilter(JPAPlugin.java:304)
2018-04-28T06:47:08.830189+00:00 app[web.1]: at play.jobs.Job.withinFilter(Job.java:201)
2018-04-28T06:47:08.830190+00:00 app[web.1]: at play.jobs.Job.call(Job.java:220)
2018-04-28T06:47:08.830192+00:00 app[web.1]: at Invocation.Job(Play!)
2018-04-28T06:47:08.830193+00:00 app[web.1]: Caused by: play.exceptions.UnexpectedException: Unexpected Error
2018-04-28T06:47:08.830194+00:00 app[web.1]: at play.libs.IO.readContent(IO.java:133)
2018-04-28T06:47:08.830196+00:00 app[web.1]: at play.templates.BaseTemplate.loadPrecompiled(BaseTemplate.java:41)
2018-04-28T06:47:08.830197+00:00 app[web.1]: ... 12 more
2018-04-28T06:47:08.830199+00:00 app[web.1]: Caused by: java.io.FileNotFoundException: File '/app/precompiled/templates/conf/data.yml' does not exist
2018-04-28T06:47:08.830200+00:00 app[web.1]: at org.apache.commons.io.FileUtils.openInputStream(FileUtils.java:292)
2018-04-28T06:47:08.830201+00:00 app[web.1]: at org.apache.commons.io.FileUtils.readFileToByteArray(FileUtils.java:1815)
2018-04-28T06:47:08.830202+00:00 app[web.1]: at play.libs.IO.readContent(IO.java:131)
2018-04-28T06:47:08.830203+00:00 app[web.1]: ... 13 more
2018-04-28T06:47:09.342593+00:00 app[web.1]: 06:47:09,342 INFO ~ Listening for HTTP on port 8575 ...
2018-04-28T06:47:09.666880+00:00 heroku[web.1]: State changed from starting to up
2018-04-28T06:47:15.508850+00:00 app[web.1]: 06:47:15,508 INFO ~ Rendering Start
2018-04-28T06:47:15.647864+00:00 heroku[router]: at=info method=GET path="/" host=todolist-edel.herokuapp.com request_id=8b963b67-6c63-43b0-b800-7571681a1328 fwd="86.44.43.185" dyno=web.1 connect=1ms service=225ms status=200 bytes=1454 protocol=https
2018-04-28T06:47:17.790158+00:00 heroku[router]: at=info method=GET path="/login" host=todolist-edel.herokuapp.com request_id=0c63d948-66b5-4d51-9c57-4fd95d8a85e1 fwd="86.44.43.185" dyno=web.1 connect=1ms service=27ms status=200 bytes=1878 protocol=https
2018-04-28T06:47:17.855407+00:00 heroku[router]: at=info method=GET path="/public/images/todo-2.jpg" host=todolist-edel.herokuapp.com request_id=31eb8230-2d5e-491a-b376-8f668de20be6 fwd="86.44.43.185" dyno=web.1 connect=1ms service=39ms status=200 bytes=173344 protocol=https
2018-04-28T06:47:25.509132+00:00 app[web.1]: 06:47:25,509 INFO ~ Attempting to authenticate with homer@simpson.com:secret
2018-04-28T06:47:25.523568+00:00 app[web.1]: 06:47:25,523 INFO ~ Authentication successful
2018-04-28T06:47:25.576685+00:00 app[web.1]: 06:47:25,576 INFO ~ Rendering Dashboard
2018-04-28T06:47:25.834299+00:00 heroku[router]: at=info method=GET path="/dashboard" host=todolist-edel.herokuapp.com request_id=daa255f3-3b12-4426-a72a-fba887e5b192 fwd="86.44.43.185" dyno=web.1 connect=2ms service=271ms status=200 bytes=2217 protocol=http
2018-04-28T06:47:25.531339+00:00 heroku[router]: at=info method=POST path="/authenticate" host=todolist-edel.herokuapp.com request_id=5befb30f-1445-4cb3-967e-4d17745e0885 fwd="86.44.43.185" dyno=web.1 connect=0ms service=46ms status=302 bytes=499 protocol=https
```

Autoscroll with output

Save

Replicates the Console visible when the app is running on localhost

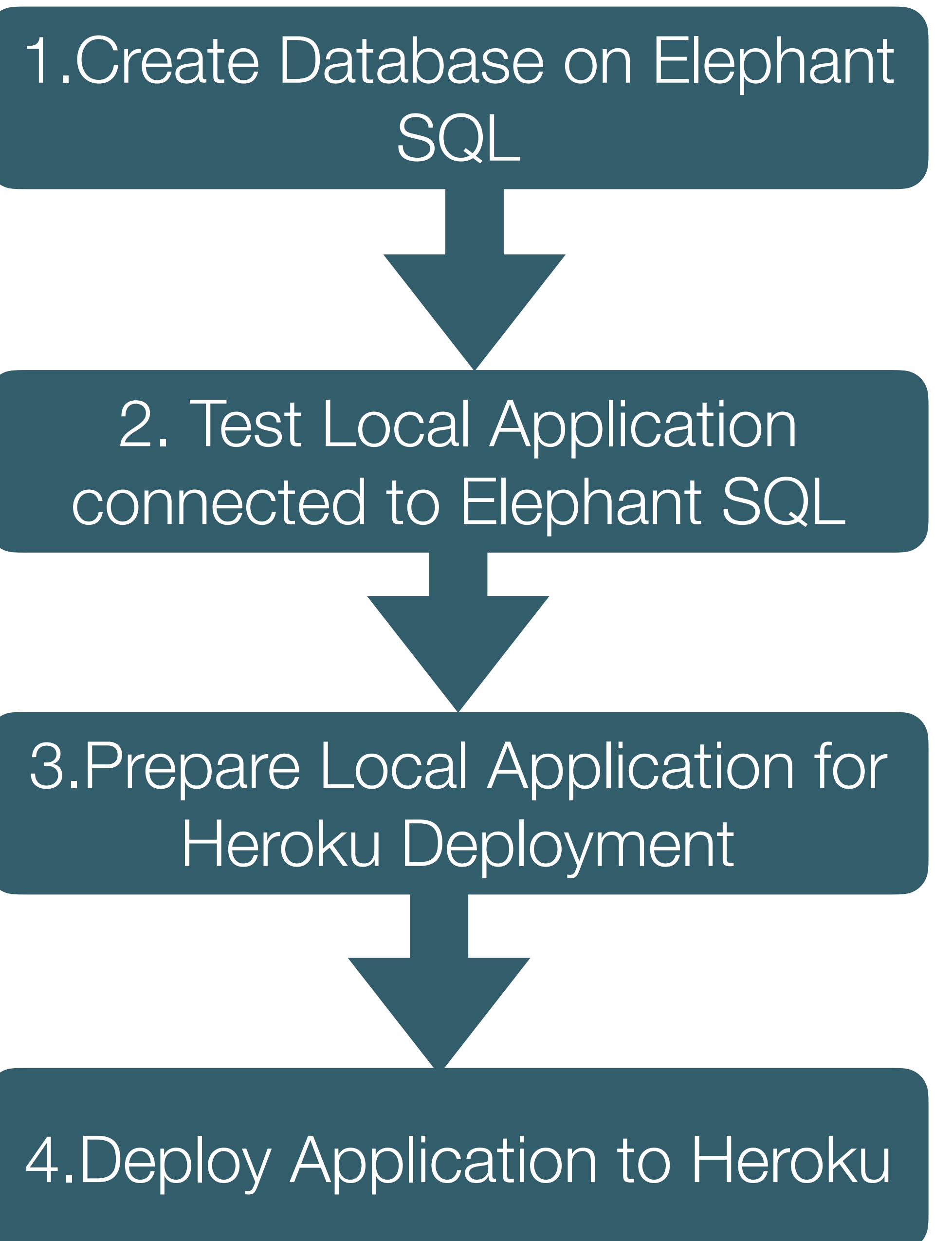
22

Three Key Configuration Parameters

- JDK Version
- Play Version
- Set Production Mode

Commit these changes:

- Push changes to Github



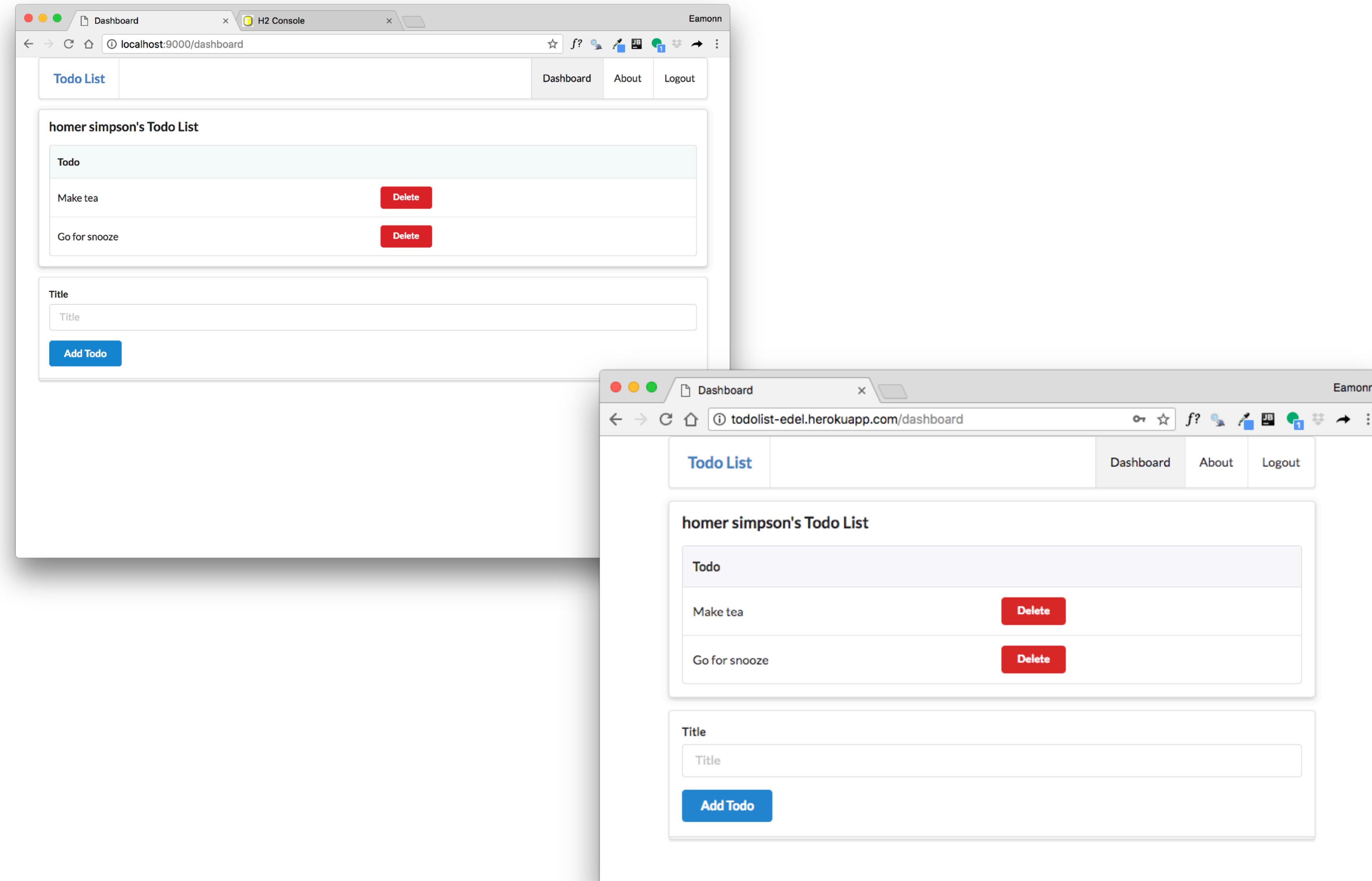
From Command Line

- Login to Heroku
- Create the Application

From Heroku Dashboard

- Link Project to Github Repo
- Deploy application
- Monitor Logs

<http://localhost:9000>



<http://todolist-edel.herokuapp.com/dashboard>

<http://localhost:9000/@db>

The screenshot shows two database management interfaces side-by-side.

Left Window (H2 Console):

- Shows the H2 database schema with tables: member, member_todo, todo, information_schema, Sequences, and Users.
- Shows the H2 version: H2 1.4.196 (2017-06-10).
- Shows the SQL statement area with the command: `SELECT * FROM "public"."member" LIMIT 100`.
- Shows the results of the query:

	id	email	firstname	lastname	password
1	4	homer@simpson.com	homer	simpson	secret
2	5	marge@simpson.com	marge	simpson	secret

Right Window (ElephantSQL SQL Browser):

- Shows the ElephantSQL logo and a user profile: HDip2021.
- Shows the SQL Browser interface with the query: `SELECT * FROM "public"."member" LIMIT 100` executed successfully.
- Shows the results of the query:

	id	email	firstname	lastname	password
1	4	homer@simpson.com	homer	simpson	secret
2	5	marge@simpson.com	marge	simpson	secret

<https://api.elephantsql.com/console/xxxxxxxx-yyyyy-zzzz/browser?>

Monitoring the Deployed app...

The image displays four screenshots illustrating the deployment and monitoring of a Heroku application named "todolist-edel".

- Heroku Dashboard (Deploy tab):** Shows the "Deploy changes" section where changes from a local Dropbox folder are being pushed to Heroku. A "Deploy" button is visible.
- Heroku Logs:** Shows the "Application Logs" for the "todolist-edel" app. The logs output the deployment process, including the command "play run --http.port=\$PORT \$PLAY_OPTS", the discovery of process types, compression, and the launch of the application. The log ends with the URL <https://todolist-edel.herokuapp.com/> deployed to Heroku.
- Todo List Application:** A screenshot of the deployed application's interface, showing a list of todos: "Make tea" and "Go for snooze". There is a "Title" input field and a "Add Todo" button.
- Network Performance Monitor:** A screenshot of the Network tab in the browser developer tools, showing a timeline of network requests. One request to "css?family=Lato:400,700,400italic,700italic&subset=latin" is highlighted.
- ElephantSQL SQL Browser:** A screenshot of the ElephantSQL interface showing the results of the query "SELECT * FROM "public"."member" LIMIT 100". The results show two rows: one for "homer@simpson.com" and one for "marge@simpson.com".