

Running Java, Play! and Scala Apps on Heroku

@_JamesWard
jamesward.com

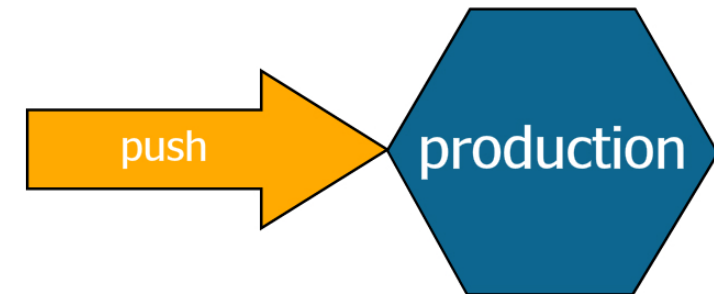
Heroku = Polyglot + PaaS + Cloud Components



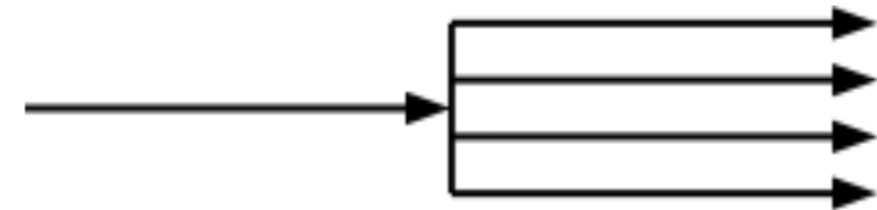
1,054,269+ Apps Running on Heroku

Cloud Application Platform

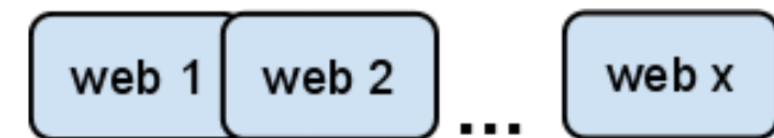
Instant Deployment



HTTP Routing / Load Balancing



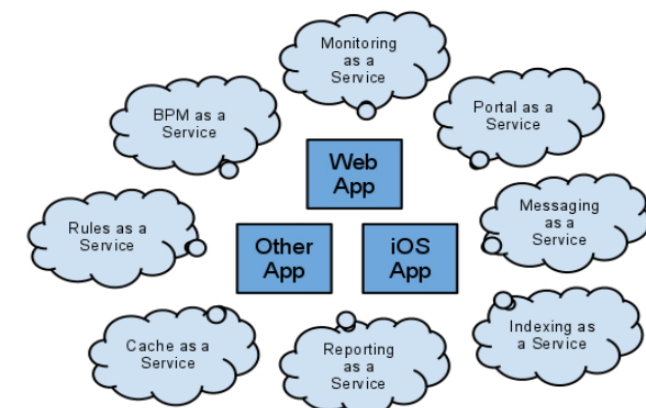
Elastic Polyglot Runtime



Management & Logging

```
2011-07-22T21:45:23+00:00 heroku[run.4]: State changed from up to complete
2011-07-22T22:10:13+00:00 heroku[slugc]: Slug compilation started
2011-07-22T22:10:25+00:00 heroku[api]: Deploy eab7965 by jw@heroku.com
2011-07-22T22:10:25+00:00 heroku[api]: Release v6 created by jw@heroku.com
```

Component as a Service Ecosystem

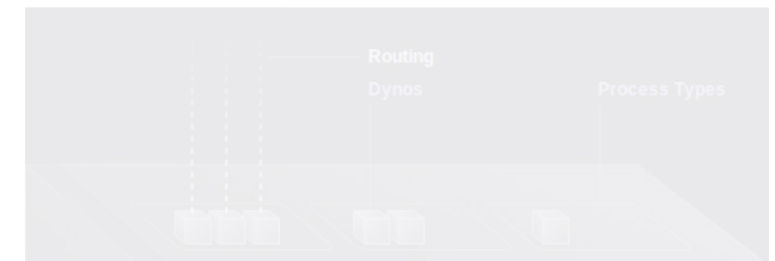


Heroku == Cloud Application Platform

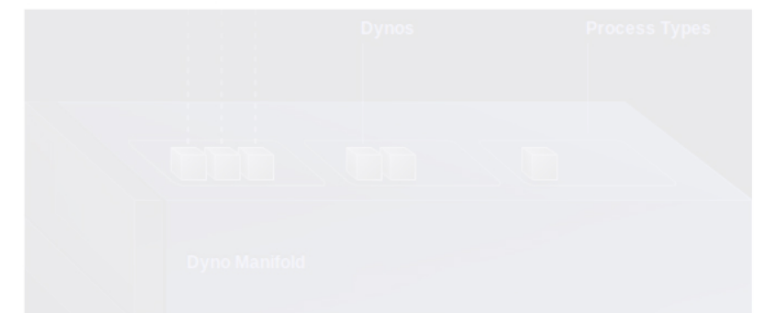
Instant Deployment

```
$ git push heroku master
```

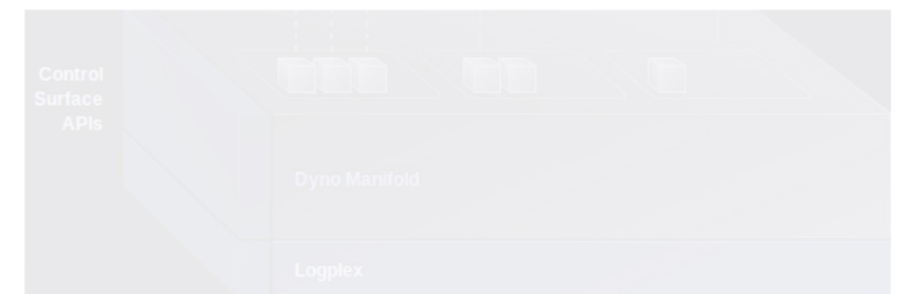
HTTP Routing / Load Balancing



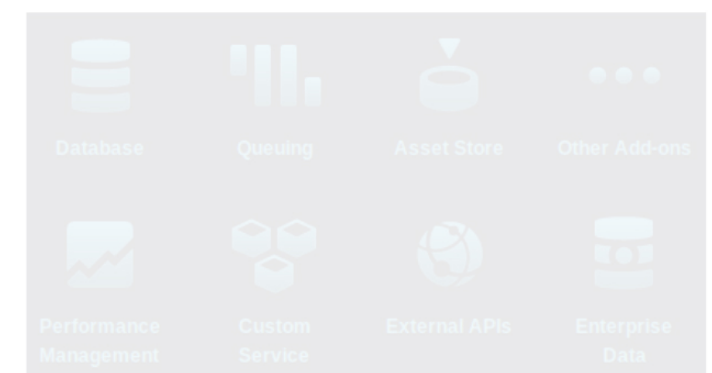
Elastic Polyglot Runtime



Management & Logging



Component as a Service Ecosystem




```
heroku create --stack cedar
```

```
git push heroku master
```

- 1) You add files to a git repo
- 2) You provision the app on Heroku (heroku create)
- 3) You upload the files to Heroku (git push heroku master)
- 4) Heroku runs the build and assembles a "slug" file
- 5) Heroku starts a "dyno"
- 6) Heroku copies the "slug" to the "dyno"
- 7) Heroku starts the web process

750 free dyno hours
per app, per month

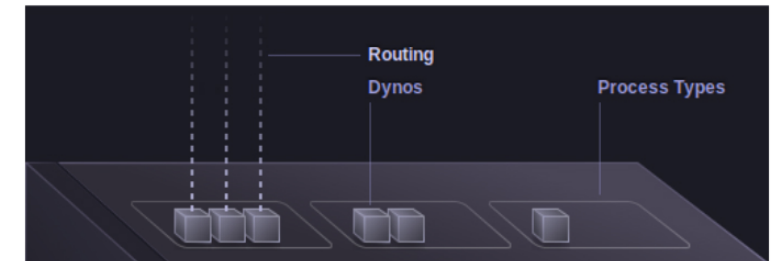
java.herokuapp.com

Heroku == Cloud Application Platform

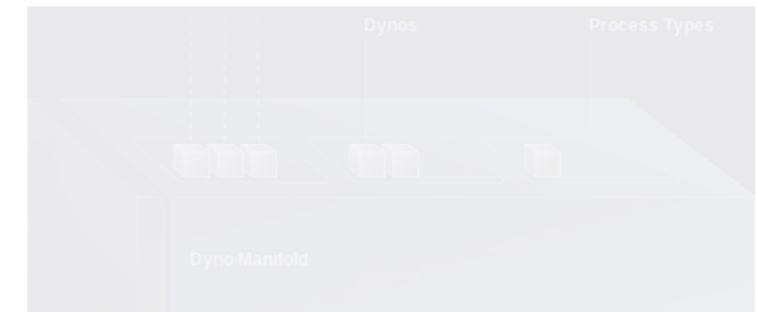
Instant Deployment

```
$ git push heroku master
```

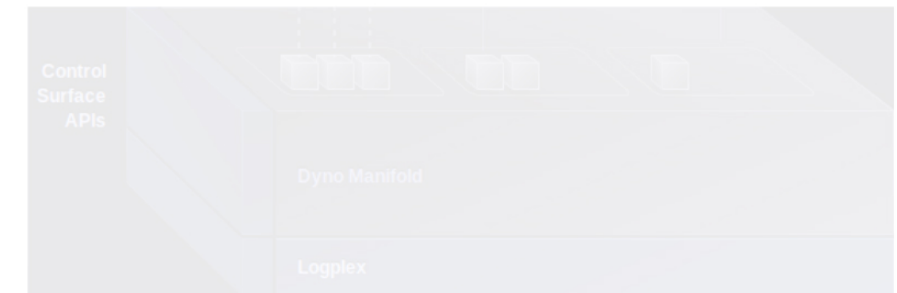
HTTP Routing / Load Balancing



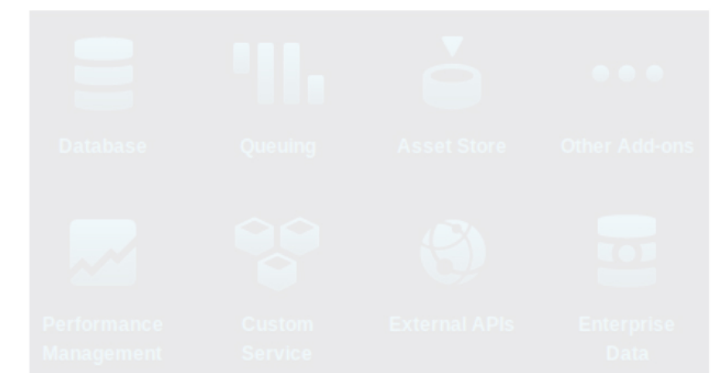
Elastic Polyglot Runtime



Management & Logging



Component as a Service Ecosystem





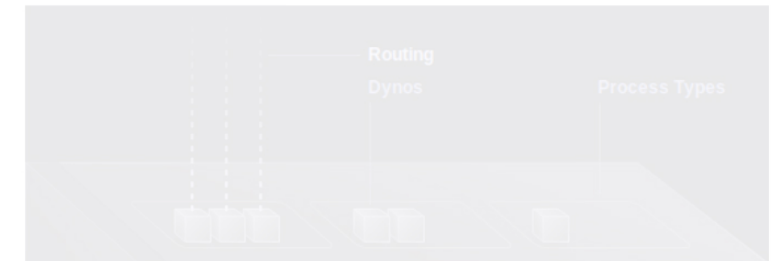
**Sticky
Sessions**

Heroku == Cloud Application Platform

Instant Deployment

```
$ git push heroku master
```

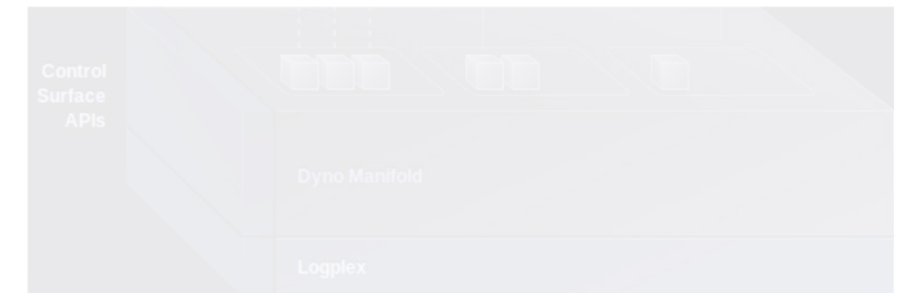
HTTP Routing / Load Balancing



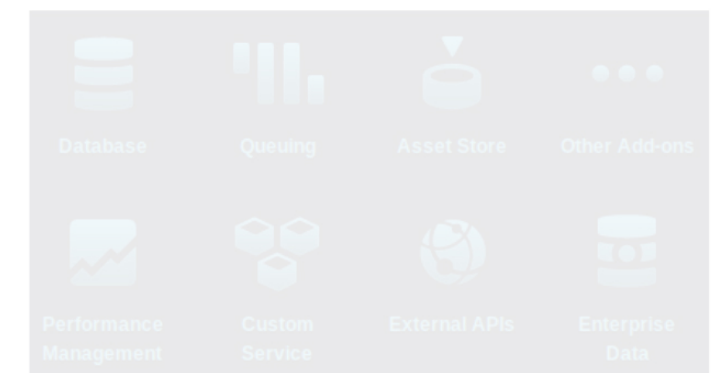
Elastic Polyglot Runtime



Management & Logging



Component as a Service Ecosystem



Create more web processes:

```
$ heroku scale web=2
```

What processes are running:

```
$ heroku ps
```


Create a Worker Process

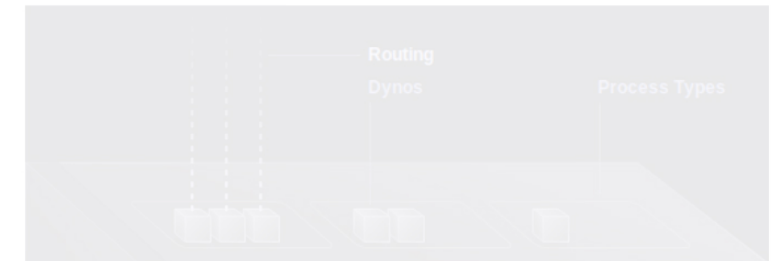
1. Create a process
2. Update the Procfile
3. Push to Heroku
4. Scale the workers

Heroku == Cloud Application Platform

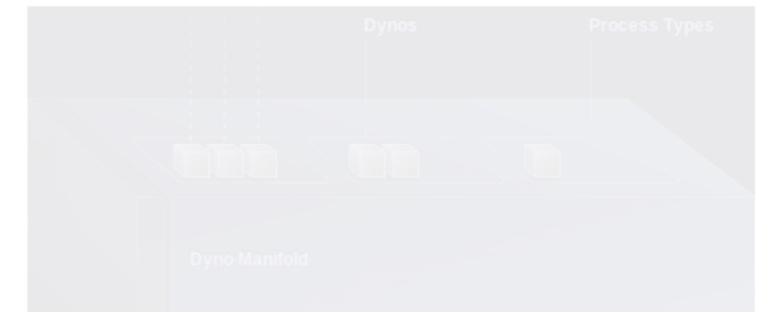
Instant Deployment

```
$ git push heroku master
```

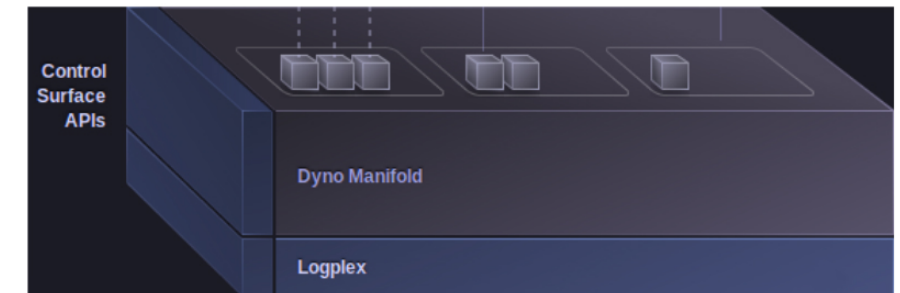
HTTP Routing / Load Balancing



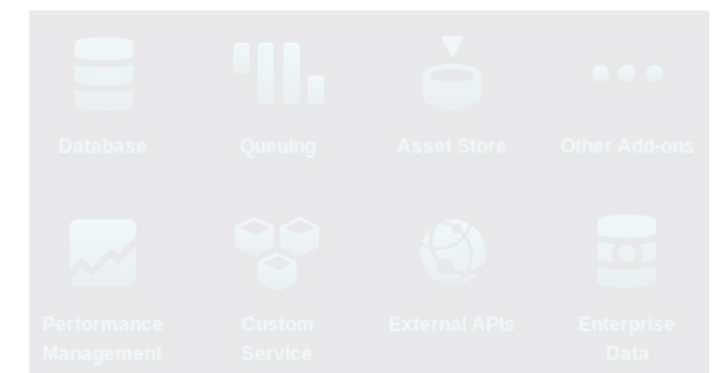
Elastic Polyglot Runtime



Management & Logging



Component as a Service Ecosystem



Check the logs:

\$ heroku logs

View environment variables:

```
$ heroku config
```

Set an environment variable:

```
$ heroku config:add MYVAR="value"
```

View releases:

\$ heroku releases

Rollback:

\$ heroku rollback

Run a process:

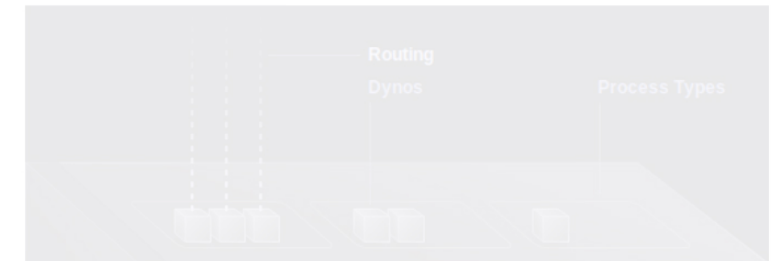
```
$ heroku run "your bash command"
```

Heroku == Cloud Application Platform

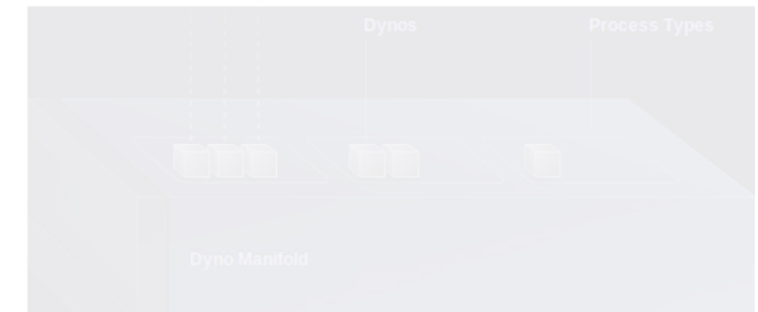
Instant Deployment

```
$ git push heroku master
```

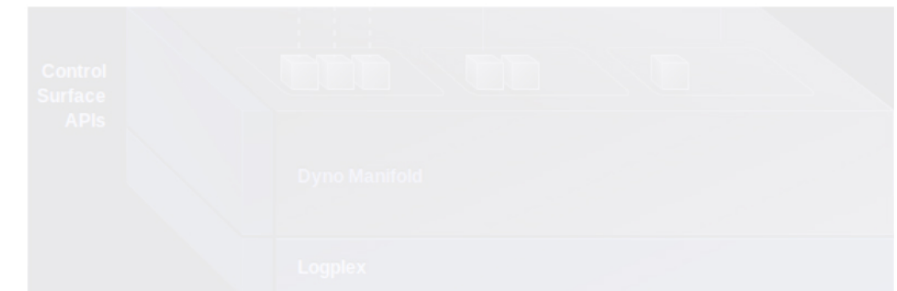
HTTP Routing / Load Balancing



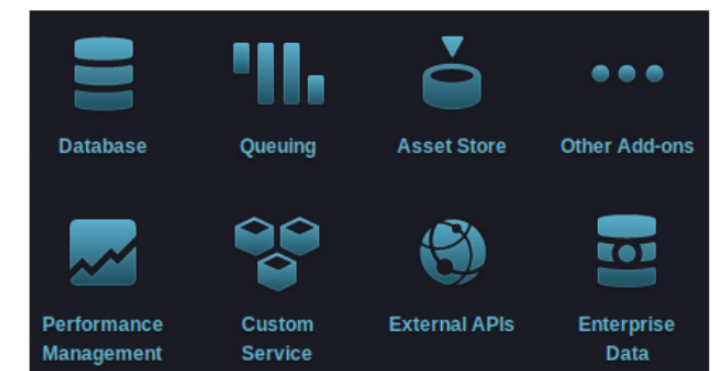
Elastic Polyglot Runtime



Management & Logging



Component as a Service Ecosystem



Use a Heroku Add-on

1. Find an Add-on: <http://addons.heroku.com>
2. Activate the Add-on
3. Use the service
4. Push to Heroku



Connect to a Database

- Free shared PostgreSQL DB
- Dedicated & managed PostgreSQL DB
- Amazon RDS MySQL & Oracle
- NoSQL Add-ons (Redis, MongoDB, etc)
- database.com
- Or do your own thing

Transform DATABASE_URL

Heroku:

```
postgres://[username]:[password]@[server]/[db-name]
```

JDBC:

```
jdbc:postgresql://[server]/[db-name]?user=[username]&password=[password]
```

Transform:

```
dbUrl = dbUrl.replaceAll("postgres://(.*) : (.*) @ (.*)",  
                          "jdbc:postgresql://$3?user=$1&password=$2");
```

Play on Heroku

- Play 1.2.3, 1.2.4, 2.0 Beta
- Java or Scala

```
$ play new foo
$ cd foo
$ git init
$ git add .
$ git commit -m init
$ heroku create -s cedar
$ git push heroku master
```

Scala on Heroku

- SBT 0.11.0
- Builds the 'stage' task

java.herokuapp.com