Erlang Deployment Options: How To Ship New Code Without Taking Your System Down

Martin Rehfeld, Wooga
@klickmich

Erlang Factory Berlin 2014

```
----------- /// ----------
```

Prelude: Update by Restart

```
----------- /// ----------
```

```
? ok     1:code- 2:run  3:slides*                 2014-12-05 08:36
```

Update by Restart   // what everybody does

+ Simple

+ Common

+ Works with all languages

+ Good tool support

- Program will lose (in-memory) state

- Update interrupts availability
  (which needs to be prevented by adding
   additional infrastructure like
   load balancers)

```
---------- /// ----------

Fugue: Hot Code Loading

---------- /// ----------
```

```
? ok      1:code- 2:run  3:slides*                    2014-12-05 08:36
```

```
|                                              |
|                                              |
|             /-\                              |
|    process (   )                             |
|             \_/                              |
|                                              |
|              |                               |
|              | runs in                       |
|              v                               |
|                                              |
|    old version m'      current version m     |
|    +----------------+  +----------------+    |
|    | a() -> b().    |  | -export([d/0]).|    |
|    |                |  |                |    |
|    | b() ->         |  | ...            |    |
|    |   ...          |  |                |    |
|    |                |  | d() ->         |    |
|    | c() -> m:d().  |  |   ...          |    |
|    +----------------+  +----------------+    |
|                                              |
|                                              |
```

```
? ok      1:code- 2:run  3:slides*          2014-12-05 08:36
```

Hot Code Loading   // Erlang speciality

+ Still relatively simple

+ Preserves in-memory state

+ Application remains available
  during update

- Erlang specific

- Limited, beware of common pitfalls

- Not much out of the box tooling,
  but good support for rolling your own

```
---------- /// ----------

Finale: Release Upgrades

---------- /// ----------
```

```
? ok     1:code- 2:run  3:slides*                    2014-12-05 08:36
```

Appup Features:

* Change internal state
* Handle module dependencies
* Change a supervisor
* Run arbitrary code during updates
* Add, remove or restart an application
* Change an application specification/ configuration

```
? ok      1:code- 2:run  3:slides*                          2014-12-05 08:36
```

```
Release Upgrades:          // Ghetto ;-)

  + All advantages of plain hot code loading

  + Covers much more edge cases


  - A lot of work to prepare AND TEST upgrades

  - You are using OTP behaviours, right?
```

```
? ok     1:code- 2:run  3:slides*                    2014-12-05 08:36
```

Examples and Slides

https://github.com/martinrehfeld/tkn-ef12014

Martin Rehfeld, Wooga
@klickmich