freevision

Best Practices to Develop Cloud Native Applications

Hello!

I am Ahmed Al Hafoudh

CEO @ freevision.co

We want …

developers to forget about
code deployment

We want ...

automated deployments

We want ...

to deploy multiple times
a day

We want …

to deploy to multiple environments

We want …

to know what is deployed where

The Twelve-Factor App

by Heroku, 2011

12 factor app

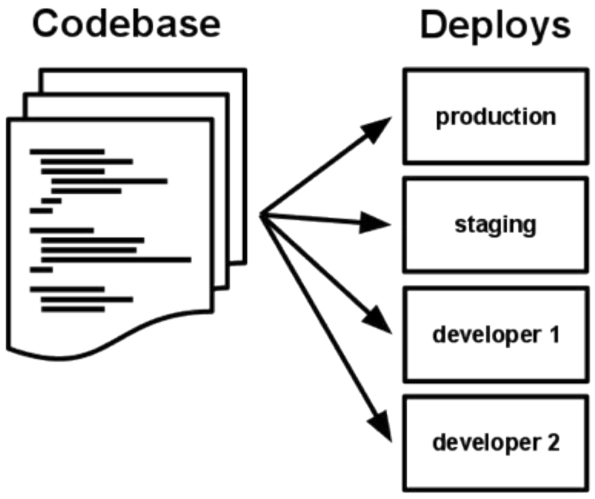
1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

12 factor app

1. **Codebase**
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

1. Codebase

- Use git (or other)
- One repository per app
- Split multiple codebases to separate repositories

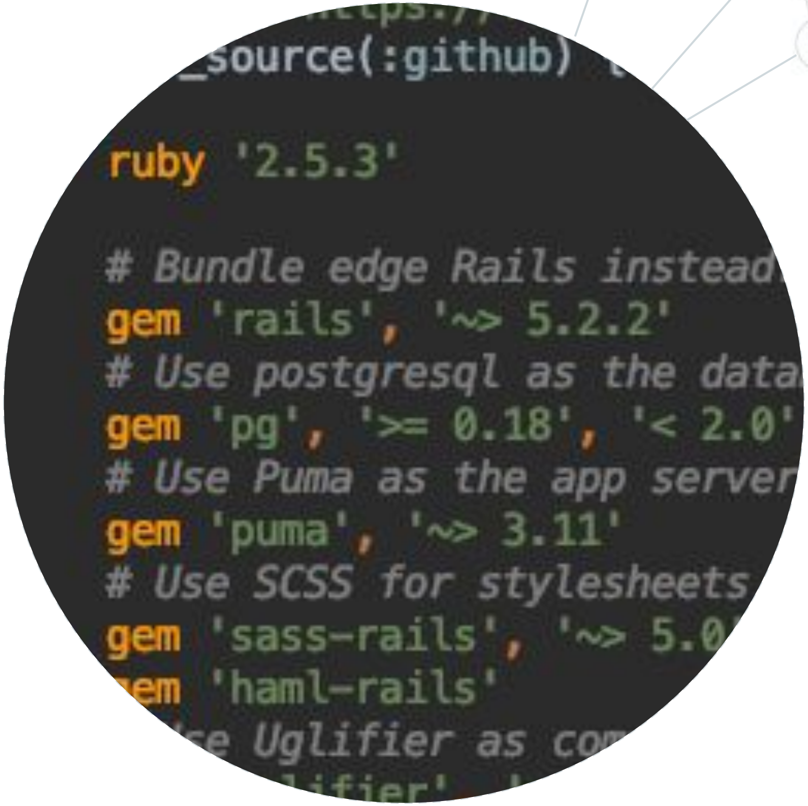


12 factor app

1. Codebase
2. **Dependencies**
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

2. Dependencies

- Don't rely on on system packages (pre-installed)
- Use package manager (bundler, yarn, pip, ...)
- Declare explicit package versions



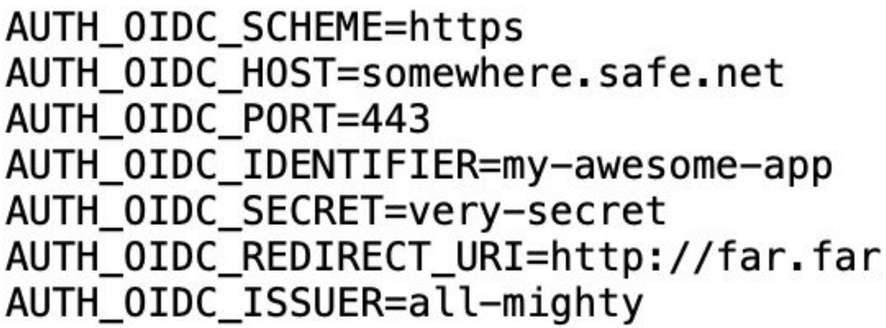
```
   ttps://
_source(:github) {

ruby '2.5.3'

# Bundle edge Rails instead
gem 'rails', '~> 5.2.2'
# Use postgresql as the data
gem 'pg', '>= 0.18', '< 2.0'
# Use Puma as the app server
gem 'puma', '~> 3.11'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5.0
em 'haml-rails'
se Uglifier as com
   lifier',
```

12 factor app

1. Codebase
2. Dependencies
3. **Config**
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

3. Config

- Don't hard code config values and credentials in code
- Don't commit credentials into the code repository
- Use environment variables (use dotenv for dev)



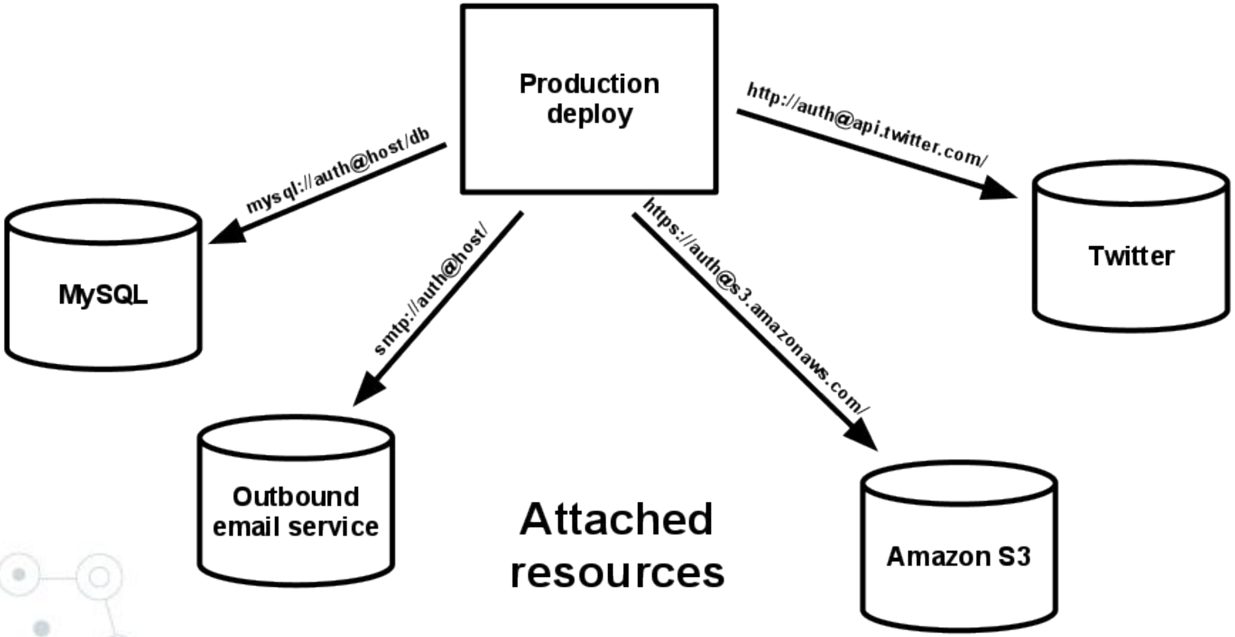
```
AUTH_OIDC_SCHEME=https
AUTH_OIDC_HOST=somewhere.safe.net
AUTH_OIDC_PORT=443
AUTH_OIDC_IDENTIFIER=my-awesome-app
AUTH_OIDC_SECRET=very-secret
AUTH_OIDC_REDIRECT_URI=http://far.far
AUTH_OIDC_ISSUER=all-mighty
```

12 factor app

1. Codebase
2. Dependencies
3. Config
4. **Backing services**
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

4. Backing services

- Reference services using single entrypoint / URL
- Treat local services as remote and use hostnames



4. Backing services

```
DATABASE_TYPE=postgres
DATABASE_USERNAME=app
DATABASE_PASSWORD=s3cr3t
DATABASE_HOST=db
DATABASE_PORT=5432
DATABASE_NAME=db1
DATABASE_SSLMODE=require
```

VS

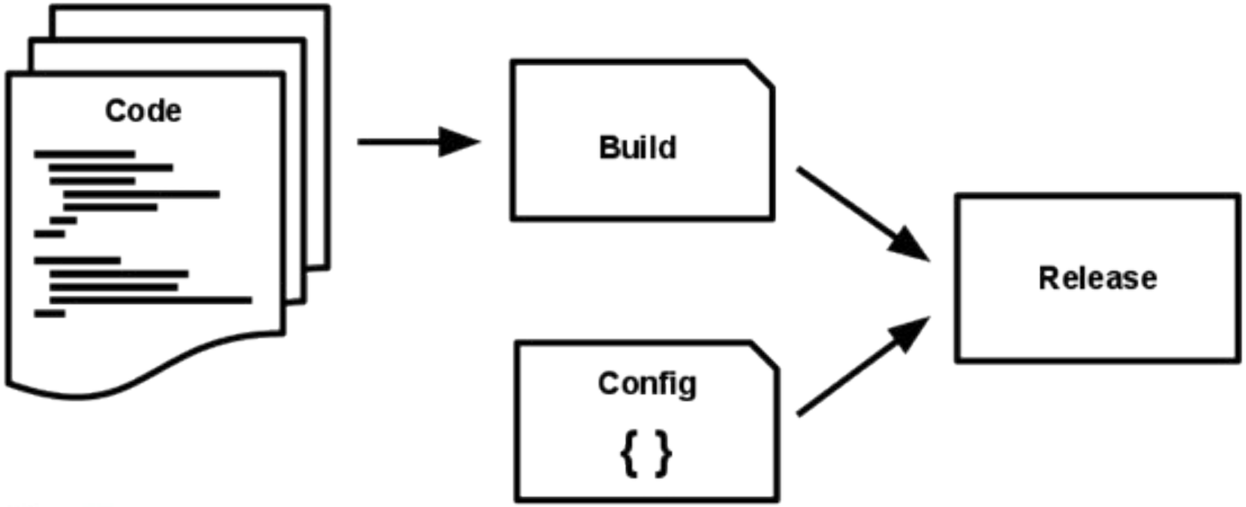
```
DATABASE_URL=
 postgresql://app:s3cr3t@db:5432/db1?sslmode=require
```

12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. **Build, release, run**
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

5. Build, release, run

- **Build** - compile and package everything
- **Release** - combine build with config and version tag
- **Run** - launch the release



12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. **Processes**
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

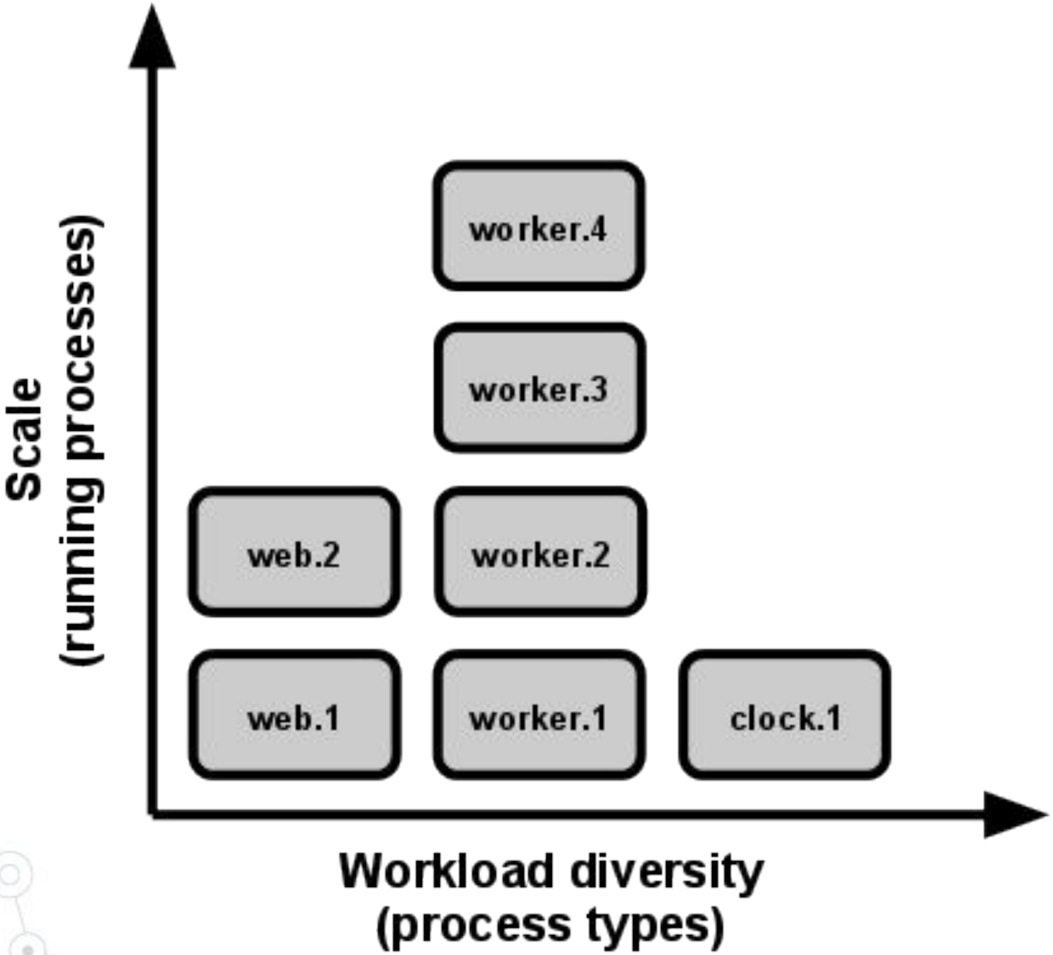
12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. **Port binding**
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. **Concurrency**
9. Disposability
10. Dev / prod parity
11. Logs
12. Admin processes

8. Concurrency



12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. **Disposability**
10. Dev / prod parity
11. Logs
12. Admin processes

9. Disposability

- Quick start
- Resilient to failure
- Graceful shutdown

12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. **Dev / prod parity**
11. Logs
12. Admin processes

10. Dev / prod parity

	dev	=	staging	=	production
✗	sqlite	≠	mysql	≠	postgresql
✓	postgresql	=	postgresql	=	postgresql

Parity ▶ Reproducibility

12 factor app

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. **Logs**
12. Admin processes

12 factor app

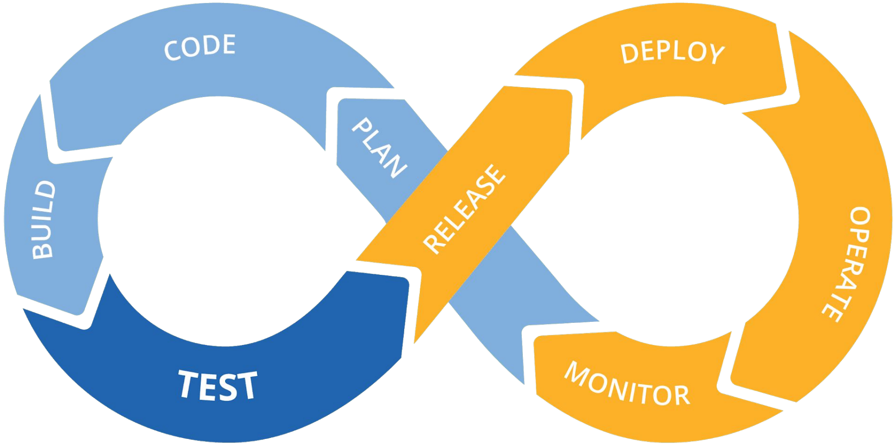
1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev / prod parity
11. Logs
12. **Admin processes**

Tips

1. Use git-flow
2. Use dotenv
3. Generate parseable logs and use ELK
4. Automate everything
5. Tag your releases
6. Expose the version tag to end user
7. Use docker
8. Try kubernetes

Summary

1. Minimize confusion, maximize productivity
2. Forget about deployments
3. Invest time to improve, improve again



Thanks!

Any questions?

alhafoudh@freevision.co