

Elixir on Kubernetes



Why?

- Horizontal and vertical autoscaling
- Your employer already uses Kubernetes

Introducing Containers

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

– [docker.com](#)

Containers

Popular flavors

- docker
- rkt

Docker

A simple example

```
FROM bitwalker/alpine-elixir:1.5.2
```

```
EXPOSE 4001
```

```
ADD mix.exs mix.lock ./
```

```
RUN mix do deps.get, deps.compile
```

```
COPY config ./config
```

```
COPY lib ./lib
```

```
COPY priv ./priv
```

```
CMD ["iex", "-S", "mix"]
```

Docker

Show & Tell

Introducing Distillery

A pure-Elixir, dependency-free implementation of release generation for Elixir projects.

– [bitwalker/distillery](#)

Distillery

Why use releases?

- Eager loading
- No development dependencies included
- Hooks and commands make it easy to run migrations

Distillery

Configuring the release

- Add distillery to your deps
- `{:distillery, "~> 1.4", runtime: false}`
- `mix release.init`
- `vi rel/config.exs`

Distillery

Show & Tell

Distillery

Bundling the release

- Goals:
 - Build anywhere
 - Release only (no dev dependencies)

Distillery & Docker

Multi-stage builds

- Introduced in Docker v17.05
- Allows sharing of files between images during build process

Distillery & Docker Show & Tell

Introducing Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

- kubernetes.io

Kubernetes

Pods

- Logical host
- One (typically) or more containers
- Share the same linux namespace
 - network -- routing table, network interfaces, ...
 - UTS -- same hostname
 - PID -- coming soon?
- Can share volumes
 - pod (not container) lifetime

Kubernetes

ReplicaSets

- Primitive resource
- Ensures desired amount of pods are always running
- Managed via Deployments

Kubernetes

Deployments

- High-level resource for configuring your application
- Responsible for pod creation via ReplicaSets
- Define scale (# of replicas)
- Define rollout strategy

Kubernetes Services

- Expose pods to external clients
 - NodePort
 - LoadBalancer
- Expose pods internally
 - ClusterIP

Kubernetes

ConfigMaps & Secrets

- Key/value pairs for configuring your application
- Secrets are base64 encoded, can hold binary data
- Can be mounted as volumes and exposed as files

Kubernetes Show & Tell

Introducing Minikube

Kubernetes, on your laptop

Minikube

Getting Started

- Start your VM
 - `minikube start`
- Set Kubernetes context to minikube
 - `kubectl config use-context minikube`
- Use Minikube Docker server
 - `eval $(minikube docker-env)`
- Access the dashboard
 - `minikube dashboard`

Minikube

Things to note

- `imagePullPolicy: Never`
 - when using local docker images

Minikube

Show & Tell

Kubernetes

Accessing your running pods

- `kubectl exec`

Elixir

Remsh & Observer

- Gain access to your running application
- NOTE: remsh is dangerous
 - Use `ssh` or `kubectl exec` for end to end encryption
 - Consider using an alternative to `epmd`

Remsh & Observer

Show & Tell

Kubernetes Logs

```
kubectl logs deployment/exonk8s -c exonk8s -f
```

Things to consider

- Graceful shutdown
 - Handling state when pods are replaced
 - Handling connections when pods are replaced

What about
distributed
Elixir?

Special Thanks

- Marko Lukša - Kubernetes in Action
- Andrew Dryga - Nebo15 / annon.api
- Rohan Relan - Polyscribe.io
- Paul Schoenfelder - distillery