



Graceful shutdown in Elixir - try not to drop the ball

Pawel Szafran



Hey, I'm Paweł

- FP pragmatist, building backends for over 10 years
- Before Elixir I coded in Java, Go, Clojure and Scala
- I did freelancing, startups and Swiss investment banks
- Now I work remotely at [Telnyx](#). We're a next-gen, API-first communications platform, that provides carrier-grade services on our global, private IP network.
We use Elixir to build e.g. services behind Voice APIs (control calls and conferences) and Fax API.
And we're hiring 





The plan

- Define the concept of graceful shutdown, and discuss when it's useful
- Go over how to implement it in Elixir and OTP
- Illustrate this with a simple demo app deployed to k8s, where we'll step-by-step implement graceful shutdown for:
 - HTTP traffic
 - async tasks
 - buffers





What is a graceful shutdown?

- Graceful/coordinated shutdown: app gets an external signal/notification to exit, and is allowed to perform some cleanup actions
- In backend systems, this is useful when we want to:
 - ship new code multiple times a day (without code reloading)
 - scale number of instances up or down (manually or with autoscaling)
 - perform maintenance activities on nodes running the serviceall without affecting users or success rate of the system!
- Recipe:
 - a. Don't take or accept new work
 - b. With a sensible timeout, finish all in-progress work, and hand off any local state



Key points

1. Test your graceful shutdown (continuously) in Staging and in Prod
2. Understand how you run and deploy services
3. It's easy to implement graceful shutdown in Elixir and OTP

▲ Let's see this in action with a DEMO APP

- Dead simple calculations API with a single /sum endpoint, which slowly adds the numbers (100ms sleep):

```
numbers=[1, 2, 3, 4], result = 10
```

- Latest Elixir, OTP, Phoenix (no HTML, no Ecto)

```
defmodule Calc.Application do
  use Application

  def start(_type, _args) do
    children = [
      CalcWeb.Telemetry,
      CalcWeb.Endpoint
    ]

    opts = [strategy: :one_for_one, name: Calc.Supervisor]
    Supervisor.start_link(children, opts)
  end
end
```

▲ Router

```
defmodule CalcWeb.Router do
  use CalcWeb, :router

  pipeline :api do
    plug :accepts, ["json"]
  end

  scope "/", CalcWeb do
    pipe_through :api

    post "/sum", CalcController, :sum
  end
end
```

▲ Controller

```
defmodule CalcWeb.CalcController do
  use CalcWeb, :controller

  def sum(conn, %{"numbers" => numbers}) when is_list(numbers) do
    result =
      numbers
      |> Enum.filter(&is_number/1)
      |> Calc.sum()

    render(conn, "result.json", result: result)
  end
end
```

▲ Calc using “slow math”

```
defmodule Calc do
  alias __MODULE__.SlowMath
  require Logger

  def sum(numbers) when is_list(numbers) do
    Logger.info("Calculating sum of #{inspect(numbers)}")
    sum = SlowMath.sum(numbers)
    Logger.info("Sum is #{sum}")
    sum
  end
end
```

```
defmodule Calc.SlowMath do
  def sum(numbers) when is_list(numbers) do
    Process.sleep(100)
    Enum.sum(numbers)
  end
end
```

Simple k8s deployment

- Packaged with Elixir Releases and multi-stage Dockerfile
- Minimal k8s deployment/service descriptors
- Deployed to 3-node k8s cluster on Digital Ocean (DO)
- 13 instances (replicas)
- Uses DO Load Balancer terminating HTTPS (later HTTP/2) with Backend Keepalive
- Real deployment may be more complex:
 - ingress controllers / edge routers
 - service mesh

▲ v2: tell k8s when pod is ready to accept HTTP traffic

```
plug CalcWeb.Plug.Health

plug Plug.RequestId
plug Plug.Telemetry, event_prefix: [:phoenix, :endpoint]
```

```
defmodule CalcWeb.Plug.Health do
  use Plug.Router

  plug(:match)
  plug(:dispatch)

  get "/health" do
    halt_with_json(conn, 200, %{status: :healthy})
  end

  match _ do
    conn
  end

  defp halt_with_json(conn, status, body) do
    conn
    ▷ put_resp_content_type("application/json")
    ▷ send_resp(status, Jason.encode!(body))
    ▷ halt()
  end
end
```

```
livenessProbe:
  exec:
    command:
      - /app/bin/calc
      - pid
  initialDelaySeconds: 2
  periodSeconds: 5
readinessProbe:
  httpGet:
    path: /health
    port: http
  initialDelaySeconds: 3
  periodSeconds: 5
```

```
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 5
    maxUnavailable: 0
```

▲ v2: ok when scaling up (3 to 13)

```
running (0m37.6s), 000/500 VUs, 9570 complete and 500 interrupted iterations
default ✘ [====>] 500 VUs 0m37.5s/5m0s

✓ 200 OK
✗ HTTP/2
↳ 0% - ✓ 0 / ✗ 9571

checks.....: 50.00% ✓ 9571 ✗ 9571
data_received.....: 34 MB 905 kB/s
data_sent.....: 5.6 MB 149 kB/s
http_req_blocked.....: avg=1.78s min=172.5ms med=1.88s max=1.92s p(90)=1.89s p(95)=1.89s
http_req_connecting.....: avg=3.28ms min=281µs med=616.55µs max=89.24ms p(90)=5.67ms p(95)=24.21ms
http_req_duration.....: avg=116.47ms min=102.84ms med=108.1ms max=420.23ms p(90)=125.41ms p(95)=164.98ms
http_req_receiving.....: avg=117.81µs min=27.79µs med=85.87µs max=15.95ms p(90)=151.41µs p(95)=192.29µs
http_req_sending.....: avg=181.62µs min=41.09µs med=113.88µs max=21.78ms p(90)=186.33µs p(95)=269.28µs
http_req_tls_handshaking ...: avg=1.78s min=134.95ms med=1.88s max=1.92s p(90)=1.89s p(95)=1.89s
http_req_waiting.....: avg=116.17ms min=102.63ms med=107.83ms max=419.67ms p(90)=124.93ms p(95)=164.69ms
http_reqs.....: 9571 254.414591/s
iteration_duration.....: avg=1.9s min=317.73ms med=1.99s max=2.3s p(90)=2s p(95)=2.02s
iterations.....: 9571 254.414591/s
vus.....: 500 min=500 max=500
vus_max.....: 500 min=500 max=500
```

▲ v2: ok when deploying... WAT?

```
running (1m11.3s), 000/500 VUs, 18032 complete and 500 interrupted iterations
default ✘ [=====] 500 VUs 1m11.2s/5m0s

✓ 200 OK
✗ HTTP/2
↳ 0% - ✓ 0 / ✗ 18032

checks.....: 50.00% ✓ 18032 ✗ 18032
data_received.....: 64 MB 898 kB/s
data_sent.....: 10 MB 146 kB/s
http_req_blocked.....: avg=1.83s min=160.66ms med=1.88s max=1.93s p(90)=1.89s p(95)=1.89s
http_req_connecting.....: avg=3.09ms min=241.01µs med=639.76µs max=101.56ms p(90)=6.86ms p(95)=12.65ms
http_req_duration.....: avg=112.35ms min=102.97ms med=108.24ms max=324.12ms p(90)=117.48ms p(95)=124.37ms
http_req_receiving.....: avg=112.85µs min=31.06µs med=86.41µs max=36.93ms p(90)=150.97µs p(95)=185.89µs
http_req_sending.....: avg=179.47µs min=37.08µs med=110.51µs max=28.61ms p(90)=181.19µs p(95)=255.28µs
http_req_tls_handshaking ...: avg=1.82s min=110.27ms med=1.88s max=1.92s p(90)=1.89s p(95)=1.89s
http_req_waiting.....: avg=112.06ms min=102.78ms med=107.96ms max=323.92ms p(90)=117.09ms p(95)=123.9ms
http_reqs.....: 18032 252.837142/s
iteration_duration.....: avg=1.94s min=401.54ms med=1.99s max=2.17s p(90)=2s p(95)=2s
iterations.....: 18032 252.837142/s
vus.....: 500 min=500 max=500
vus_max.....: 500 min=500 max=500
```

▲ v2: all pods terminating... SIGTERM ignored

```
<@graceful-shutdown-demo>→[ master]>→ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
calc-7478555458-2td9t  1/1     Running   0          25s
calc-7478555458-8mn8k  1/1     Running   0          22s
calc-7478555458-b6fq6  1/1     Running   0          32s
calc-7478555458-dr8x9  1/1     Running   0          17s
calc-7478555458-frbmb  1/1     Running   0          25s
calc-7478555458-frdf2  1/1     Running   0          32s
calc-7478555458-gpv7m  1/1     Running   0          19s
calc-7478555458-k5zhr  1/1     Running   0          18s
calc-7478555458-ljn2z  1/1     Running   0          32s
calc-7478555458-pcmz4  1/1     Running   0          26s
calc-7478555458-pxdx2  1/1     Running   0          26s
calc-7478555458-qvnll  1/1     Running   0          32s
calc-7478555458-vrw9m  1/1     Running   0          32s
calc-8cb555bc-2qj28    1/1     Terminating   0          3m27s
calc-8cb555bc-595cw    1/1     Terminating   0          3m30s
calc-8cb555bc-5z4vx    1/1     Terminating   0          3m31s
calc-8cb555bc-7psnr    1/1     Terminating   0          3m27s
calc-8cb555bc-dbwtp    1/1     Terminating   0          3m39s
calc-8cb555bc-h2c7v    1/1     Terminating   0          3m39s
calc-8cb555bc-jwwst    1/1     Terminating   0          3m39s
calc-8cb555bc-mb8sb    1/1     Terminating   0          3m39s
calc-8cb555bc-nglvw    1/1     Terminating   0          3m29s
calc-8cb555bc-pv7tq    1/1     Terminating   0          3m33s
calc-8cb555bc-qh7gd    1/1     Terminating   0          3m40s
calc-8cb555bc-tzlm8    1/1     Terminating   0          3m22s
calc-8cb555bc-vv5ng    1/1     Terminating   0          3m32s
<@graceful-shutdown-demo>→[ master]>→
```

▲ v2: Bash doesn't forward OS signals to child process

```
FROM alpine:${ALPINE_VERSION} as run

RUN apk add --update --no-cache bash openssl

WORKDIR /app

COPY --from=build /build/_build/prod/rel/calc ./
COPY bin/start.sh start.sh|
RUN chmod 755 start.sh

RUN addgroup -S app && adduser -S app -G app
RUN chown -R app:app /app
USER app

EXPOSE 4000

CMD ["/app/start.sh"]
```

```
#!/usr/bin/env bash

# E.g. run data migration
# /app/bin/calc eval "Calc.Release.migrate()"

/app/bin/calc start
|
```

▲ v2: test with Docker... no SIGTERM

```
docker run -it --rm --name calc -p 4000:4000 calc
23:13:38.875 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
23:13:38.876 [info] Access CalcWeb.Endpoint at http://localhost:4000
make: *** [Makefile:106: docker-local-server] Error 137
<@graceful-shutdown-demo>-<./ master>-2→ □
<@graceful-shutdown-demo>-<./ master>-> docker stop calc
calc
<@graceful-shutdown-demo>-<./ master>-> █
```

▲ v3: use exec to get SIGTERM (OTP 19.3+)

```
#!/usr/bin/env bash

# E.g. run data migration
# /app/bin/calc eval "Calc.Release.migrate()"

exec /app/bin/calc start
```

```
docker run -it --rm --name calc -p 4000:4000 calc
23:15:34.623 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
23:15:34.623 [info] Access CalcWeb.Endpoint at http://localhost:4000
23:15:37.118 [info] SIGTERM received - shutting down
```

```
<@graceful-shutdown-demo>-<./ master>-<±>-> []
<@graceful-shutdown-demo>-<./ master>-> docker stop calc
calc
<@graceful-shutdown-demo>-<./ master>-> docker stop calc
calc
<@graceful-shutdown-demo>-<./ master>-<±>-> []
```

▲ v3: or directly call releases cmd

```
CMD ["/app/bin/calc", "start"]
```

```
docker run -it --rm --name calc -p 4000:4000 calc
23:19:41.359 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
23:19:41.359 [info] Access CalcWeb.Endpoint at http://localhost:4000
23:19:43.776 [info] SIGTERM received - shutting down
```

```
<@graceful-shutdown-demo>→[ master>→±-> ]
```

```
<@graceful-shutdown-demo>→[ master>→±-> docker stop calc
calc
```

```
<@graceful-shutdown-demo>→[ master>→±-> ]
```

Setup task like DB migrations should be done using k8s Init Containers or k8s Tasks

 v3: errors when deploying... noooooooo

▲ v4: if this talk was half a year ago

```
ERRO[0086] Not 200 OK: {"status":502,"error":""}
ERRO[0088] Not 200 OK: {"status":502,"error":""}
WARN[0088] Request Failed
ERRO[0088] Not 200 OK: {"status":0,"error":"EOF"}
ERRO[0088] Not 200 OK: {"status":502,"error":""}
ERRO[0088] Not 200 OK: {"status":502,"error":""}
ERRO[0088] Not 200 OK: {"status":502,"error":""}

running (1m56.8s), 000/500 VUs, 29310 complete and 500 interrupted iterations
default x [=====] 500 VUs 1m56.7s/5m0s

x 200 OK
↳ 99% - ✓ 29264 / x 46
x HTTP/2
↳ 0% - ✓ 0 / x 29310

checks.....: 49.92% ✓ 29264 x 29356
data_received.....: 104 MB 891 kB/s
data_sent.....: 17 MB 145 kB/s
http_req_blocked.....: avg=1.76s min=144.2ms med=1.82s max=2s p(90)=1.86s p(95)=1.87s
http_req_connecting.....: avg=3.51ms min=265.29µs med=1.08ms max=110.48ms p(90)=7.85ms p(95)=14.45ms
http_req_duration.....: avg=204.05ms min=18.38ms med=153.19ms max=1.47s p(90)=357.84ms p(95)=472.18ms
http_req_receiving.....: avg=69.12µs min=0s med=55.84µs max=13.24ms p(90)=95.25µs p(95)=112.69µs
http_req_sending.....: avg=98.44µs min=29.07µs med=75.97µs max=16.29ms p(90)=114.68µs p(95)=133.82µs
http_req_tls_handshaking ..: avg=1.76s min=83.64ms med=1.81s max=2s p(90)=1.86s p(95)=1.87s
http_req_waiting.....: avg=203.88ms min=18.13ms med=153.05ms max=1.47s p(90)=357.71ms p(95)=472.03ms
http_reqs.....: 29310 250.906918/s
iteration_duration.....: avg=1.97s min=518.68ms med=1.97s max=3.26s p(90)=2.12s p(95)=2.24s
iterations.....: 29310 250.906918/s
vus.....: 500 min=500 max=500
vus_max.....: 500 min=500 max=500
```

▲ v4: simulate Phoenix before 1.5

- Phoenix 1.5 comes by default with `Plug.Cowboy.Drainer`, which drains `cowboy` HTTP connections at shutdown (calling `ranch`)
- To simulate behavior before that version, we can:

```
config :calc, CalcWeb.Endpoint,  
  http: [:inet6, port: port],  
  url: [host: "localhost", port: port],  
  render_errors: [view: CalcWeb.ErrorView, accepts: ~w(json), layout: false],  
  drainer: false
```

▲ v5: connections are drained in v3, so why errors?

```
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.327997466Z 19:57:08.268 request_id=FjFeTS3dUz80eEAAAFrR [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.339676146Z 19:57:08.329 [info] SIGTERM received - shutting down
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.339738971Z
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360280621Z 19:57:08.337 request_id=FjFeTTIAoN10508AAFRS [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360469531Z 19:57:08.338 request_id=FjFeTTIFth0Q68UAAFRi [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360611936Z 19:57:08.338 request_id=FjFeTTINeR1KfV0AAFRy [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360637484Z 19:57:08.338 request_id=FjFeTTIRLt_YA6gAAFSC [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360644742Z 19:57:08.346 request_id=FjFeTS3dUz80eEAAAFrR [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360685496Z 19:57:08.346 request_id=FjFeTSvSWOSKOxQAAFx [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360694159Z 19:57:08.357 request_id=FjFeTTIAoN10508AAFRS [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360704756Z 19:57:08.357 request_id=FjFeTTIFth0Q68UAAFRi [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360713621Z 19:57:08.357 request_id=FjFeTTINeR1KfV0AAFRy [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360723180Z 19:57:08.357 request_id=FjFeTTIRLt_YA6gAAFSC [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.360732473Z 19:57:08.357 request_id=FjFeTSxRCLCJ6FMAAFrB [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.368740425Z 19:57:08.358 request_id=FjFeTTM-5Md4HuCAFSS [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.368788192Z 19:57:08.358 request_id=FjFeTSxRCLCJ6FMAAFrB [info] Sent 200 in 116ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.368799588Z 19:57:08.359 request_id=FjFeTTM-5Md4HuCAFSS [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.368944412Z 19:57:08.368 request_id=FjFeTSyUovtGXbMAAFRC [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.369377434Z 19:57:08.368 request_id=FjFeTSyUovtGXbMAAFRC [info] Sent 200 in 122ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.383045683Z 19:57:08.378 request_id=FjFeTSvSWOSKOxQAAFx [info] Sent 200 in 112ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491462472Z 19:57:08.467 request_id=FjFeTTIAoN10508AAFRS [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491555727Z 19:57:08.467 request_id=FjFeTTIFth0Q68UAAFRi [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491566093Z 19:57:08.468 request_id=FjFeTTINeR1KfV0AAFRy [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491572648Z 19:57:08.468 request_id=FjFeTTIRLt_YA6gAAFSC [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491579071Z 19:57:08.468 request_id=FjFeTTM-5Md4HuCAFSS [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491585574Z 19:57:08.471 request_id=FjFeTTIFth0Q68UAAFRi [info] Sent 200 in 133ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491592028Z 19:57:08.472 request_id=FjFeTS3dUz80eEAAAFrR [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491598197Z 19:57:08.472 request_id=FjFeTS3dUz80eEAAAFrR [info] Sent 200 in 204ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491604249Z 19:57:08.472 request_id=FjFeTTINeR1KfV0AAFRy [info] Sent 200 in 134ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491610307Z 19:57:08.472 request_id=FjFeTTIRLt_YA6gAAFSC [info] Sent 200 in 134ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.491616539Z 19:57:08.472 request_id=FjFeTTM-5Md4HuCAFSS [info] Sent 200 in 114ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.521313014Z 19:57:08.478 request_id=FjFeTTIAoN10508AAFRS [info] Sent 200 in 130ms
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.547760954Z 19:57:08.497 request_id=FjFeTTUDSug0BtAAAFSi [info] POST /sum
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.547807861Z 19:57:08.527 request_id=FjFeTTUDSug0BtAAAFSi [info] Calculating sum of [1, 2, 3, 4]
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.636266472Z 19:57:08.634 request_id=FjFeTTUDSug0BtAAAFSi [info] Sum is 10
calc-b8b6868bc-2rgtg calc 2020-09-03T19:57:08.636294642Z 19:57:08.635 request_id=FjFeTTUDSug0BtAAAFSi [info] Sent 200 in 137ms
- calc-b8b6868bc-2rgtg
```

▲ v5: pre-stop-sleep

- Removing Pods from Endpoints is async
- Solutions: give k8s time to remove Pod from Endpoints, before Elixir gets SIGTERM

```
lifecycle:  
  preStop:  
    exec:  
      command: ["sh", "-c", "sleep 5"]
```


▲ v6: async work using Tasks

```
defmodule Calc do
  alias __MODULE__.SlowMath
  require Logger

  def sum(numbers) when is_list(numbers) do
    Logger.info("Calculating sum of #{inspect(numbers)}")
    sum = SlowMath.sum(numbers)
    Logger.info("Sum is #{sum}")
    Calc.Log.publish_async(:sum, numbers, sum)
    sum
  end
end
```

▲ v6: send every calculation to InfluxDB over HTTP after 5s

```
defmodule Calc.Log do
  require Logger

  def publish_async(op, numbers, result) do
    Logger.info("Logging operation... ")
    numbers = length(numbers)

    Task.start(fn →
      publish(op, numbers, result)
    end)
  end

  defp publish(op, numbers, result) do
    Process.sleep(5_000)
    write_to_influxdb("operation,name=#{op} numbers=#{numbers},result=#{result}")
  end

  defp write_to_influxdb(log) do
    Finch.build(:post, influxdb_url("/write?db=calc"), [], log)
    ▷ Finch.request(CalcFinch)
    ▷ parse_response()
  end
end
```

v6: bump cluster

- Nodes: 3 to 7
- Max deployment surge: 5 to 3
- `livenessProbe.initialDelaySeconds`: 2 to 10
- `readinessProbe.initialDelaySeconds`: 3 to 20

Graceful shutdown in Elixir

- SIGTERM translates to `System.stop/0`
- `System.stop/0` stops all applications in the opposite order they were started
- Application stops in 3 steps:
 - `prep_stop/1` callback
 - terminate top-level supervisor
 - `stop/1` callback

Application callbacks

```
defmodule Calc.Application do
  use Application
  require Logger

  def start(_type, _args) do
    Logger.info("App starting ... ")

    children = [
      {Finch, name: CalcFinch},
      CalcWeb.Telemetry,
      CalcWeb.Endpoint
    ]

    opts = [strategy: :one_for_one, name: Calc.Supervisor]
    supervisor = Supervisor.start_link(children, opts)
    Logger.info("App started")
    supervisor
  end

  def prep_stop(_state) do
    Logger.info("App stopping ... ")
  end

  def stop(_state) do
    Logger.info("App stopped")
  end
end
```

▲ IEx: Ctrl+C

```
Compiling 1 file (.ex)
21:11:02.147 [info] App starting ...
21:11:02.192 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
21:11:02.197 [info] Access CalcWeb.Endpoint at http://localhost:4000
21:11:02.197 [info] App started
Interactive Elixir (1.10.4) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> 21:11:20.318 request_id=FjJHixMeWpBdZJwAAAEi [info] POST /sum
21:11:20.334 request_id=FjJHixMeWpBdZJwAAAEi [debug] Processing with CalcWeb.CalcController.sum/2
  Parameters: %{"numbers" => [1, 2, 3, 4]}
  Pipelines: [:api]
21:11:20.336 request_id=FjJHixMeWpBdZJwAAAEi [info] Calculating sum of [1, 2, 3, 4]
21:11:20.437 request_id=FjJHixMeWpBdZJwAAAEi [info] Sum is 10
21:11:20.437 request_id=FjJHixMeWpBdZJwAAAEi [info] Logging operation ...
21:11:20.448 request_id=FjJHixMeWpBdZJwAAAEi [info] Sent 200 in 130ms

BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
       (l)oaded (v)ersion (k)ill (D)b-tables (d)istribution
^C ↵
<@graceful-shutdown-demo>-<I master>-<±>-130→ □
<@graceful-shutdown-demo>-<I master>-> make local-test
curl -X POST http://localhost:4000/sum \
      -H 'Content-Type: application/json' \
      -d '{"numbers": [1, 2, 3, 4]}'
{"result":10} ↵
<@graceful-shutdown-demo>-<I master>-<±>-> □
```

▲ IEx: System.stop()

```
21:13:14.771 [info] App starting ...
21:13:14.808 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
21:13:14.812 [info] Access CalcWeb.Endpoint at http://localhost:4000
21:13:14.812 [info] App started
Interactive Elixir (1.10.4) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> 21:13:25.117 request_id=FjJHqCHG0Rjfov8AAC [info] POST /sum
21:13:25.131 request_id=FjJHqCHG0Rjfov8AAC [debug] Processing with CalcWeb.CalcController.sum/2
  Parameters: %{"numbers" => [1, 2, 3, 4]}
  Pipelines: [:api]
21:13:25.133 request_id=FjJHqCHG0Rjfov8AAC [info] Calculating sum of [1, 2, 3, 4]
21:13:25.234 request_id=FjJHqCHG0Rjfov8AAC [info] Sum is 10
21:13:25.234 request_id=FjJHqCHG0Rjfov8AAC [info] Logging operation ...
21:13:25.246 request_id=FjJHqCHG0Rjfov8AAC [info] Sent 200 in 129ms
System.stop()
:ok
21:13:25.958 [info] App stopping ...
iex(2)> 21:13:25.959 [info] App stopped
<@graceful-shutdown-demo>-<./ master>-<±>-> █
<@graceful-shutdown-demo>-<./ master>-<±>-> make local-test
curl -X POST http://localhost:4000/sum \
      -H 'Content-Type: application/json' \
      -d '{"numbers": [1, 2, 3, 4]}'
{"result":10}█
<@graceful-shutdown-demo>-<./ master>-<±>-> █
```

Supervisor

- Starts children in the order they are defined (top down)
- Terminates children in the opposite order they are defined (bottom up)
- Start:
 - calls function under :start key in the child spec (typically start_link/1)
 - child process usually starts its work by executing the init/1
- Shutdown:
 - sends the child process a :shutdown exit signal
 - awaits with a timeout for the child process to terminate
 - timeout is defined in the child spec: defaults to 5s for workers and to infinity for supervisors
 - if the child process doesn't terminate in time, it sends a :kill signal

Child process

- If child is not trapping exits, it shuts down immediately on :shutdown exit signal
- If child is trapping exits, then the terminate/2 callback is called
- *Note: exit signal is just a message to a process, and it waits in the same queue with other messages!*

How to implement graceful shutdown in Elixir

1. Put important processes under supervision tree
2. Define proper shutdown timeouts (do timeout math)
3. Trap exits
4. Implement `terminate/2` callback

▲ Example

```
def start(_type, _args) do
  Logger.info("App starting ... ")

  children = [
    {Finch, name: CalcFinch},
    CalcWeb.Telemetry,
    CalcWeb.Endpoint,
    Calc.Foo,
    Calc.Bar
  ]
```

```
defmodule Calc.Foo do
  use GenServer
  require Logger

  def start_link(_opts) do
    GenServer.start_link(__MODULE__, :ok, name: __MODULE__)
  end

  @impl true
  def init(:ok) do
    Process.flag(:trap_exit, true)
    Logger.info("Foo started")
    {:ok, nil}
  end

  @impl true
  def terminate(reason, _state) do
    Logger.info("Foo terminating with reason: #{reason}")
    Process.sleep(2_000) ← 7s for Bar
    Logger.info("Foo terminated")
  end
end
```

▲ Shutdown sequence

```
<@graceful-shutdown-demo>-<./ master>-<±>-> make local-server
exec iex -S mix phx.server
Erlang/OTP 23 [erts-11.0.3] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [hipe]

10:54:25.796 [info] App starting ...
10:54:25.834 [info] Running CalcWeb.Endpoint with cowboy 2.8.0 at :::4000 (http)
10:54:25.838 [info] Access CalcWeb.Endpoint at http://localhost:4000
10:54:25.838 [info] Foo started
10:54:25.838 [info] Bar started
10:54:25.838 [info] App started
Interactive Elixir (1.10.4) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> System.stop
:ok
10:54:28.208 [info] App stopping ...
10:54:28.208 [info] Bar terminating with reason: shutdown
iex(2)> 10:54:33.209 [info] Foo terminating with reason: shutdown
10:54:35.209 [info] Foo terminated
10:54:35.210 [info] App stopped
<@graceful-shutdown-demo>-<./ master>-<±>-> █
```

Fix timeout for Bar

```
defmodule Calc.Bar do
  use GenServer, shutdown: 10_000
  require Logger
```

```
iex(2)> Calc.Foo.child_spec([])
%{id: Calc.Foo, start: {Calc.Foo, :start_link, []}}
iex(3)> Calc.Bar.child_spec([])
%{id: Calc.Bar, shutdown: 10000, start: {Calc.Bar, :start_link, []}}
iex(4)> System.stop
:ok
11:15:48.064 [info] App stopping ...
11:15:48.064 [info] Bar terminating with reason: shutdown
iex(5)> 11:15:55.064 [info] Bar terminated
11:15:55.064 [info] Foo terminating with reason: shutdown
11:15:57.065 [info] Foo terminated
11:15:57.067 [info] App stopped
<@graceful-shutdown-demo>-<L master>-<±>->
```

▲ And if you don't control the source code...

```
iex(3)> Calc.Foo.child_spec([])
%{id: Calc.Foo, start: {Calc.Foo, :start_link, []}}
iex(4)> Supervisor.child_spec(Calc.Foo, shutdown: 10_000)
%{id: Calc.Foo, shutdown: 10000, start: {Calc.Foo, :start_link, []}}
iex(5)> █
```

▲ Exit signals are just a messages

```
def ping(id), do: GenServer.cast(__MODULE__, {:ping, id})\n\n@impl true\ndef handle_cast({:ping, id}, state) do\n  Logger.info("Foo processing ping-#{id} ...")\n  Process.sleep(1_000)\n  Logger.info("Foo processed ping-#{id}")\n  {:noreply, state}\nend
```

```
@impl true\ndef init(:ok) do\n  Process.flag(:trap_exit, true)\n  Logger.info("Bar started")\n  {:ok, nil, {:continue, :ping_foo}}\nend\n\n@impl true\ndef handle_continue(:ping_foo, state) do\n  for id <- 1..100, do: Calc.Foo.ping(id)\n  {:noreply, state}\nend
```

```
System.stop()\n11:59:26.805 [info] App stopping ...\n:ok\n11:59:26.805 [info] Bar terminating with reason: shutdown\niex(2)> 11:59:27.559 [info] Foo processed ping-8\n11:59:27.559 [info] Foo processing ping-9 ...\n11:59:28.560 [info] Foo processed ping-9\n11:59:28.560 [info] Foo processing ping-10 ...\n11:59:29.561 [info] Foo processed ping-10\n11:59:29.561 [info] Foo processing ping-11 ...\n11:59:30.562 [info] Foo processed ping-11\n11:59:30.562 [info] Foo processing ping-12 ...\n11:59:31.563 [info] Foo processed ping-12\n11:59:31.563 [info] Foo processing ping-13 ...\n11:59:32.563 [info] Foo processed ping-13\n11:59:32.564 [info] Foo processing ping-14 ...\n11:59:33.565 [info] Foo processed ping-14\n11:59:33.565 [info] Foo processing ping-15 ...\n11:59:33.806 [info] Bar terminated\n11:59:34.566 [info] Foo processed ping-15\n11:59:34.566 [info] Foo processing ping-16 ...\n11:59:35.567 [info] Foo processed ping-16\n11:59:35.567 [info] Foo processing ping-17 ...\n11:59:36.568 [info] Foo processed ping-17\n11:59:36.568 [info] Foo processing ping-18 ...\n11:59:37.569 [info] Foo processed ping-18\n11:59:37.569 [info] Foo processing ping-19 ...\n11:59:38.570 [info] Foo processed ping-19\n11:59:38.570 [info] Foo processing ping-20 ...\n11:59:38.808 [info] App stopped\n<@graceful-shutdown-demo>-<./master>-<1>->
```

▲ Exit signals are just a messages

```
Calc.Foo ▷ Process.whereis() ▷ Process.info(:messages) ▷ elem(1) ▷ Enum.take(-5)
```

```
12:22:23.718 [info] Bar terminated
Calc.Foo ▷ Process.whereis() ▷ Process.info(:messages) ▷ elem(1) ▷
g-21
12:22:24.635 [info] Foo processing ping-22 ...

[
  {:$gen_cast, {:ping, 97}},
  {:$gen_cast, {:ping, 98}},
  {:$gen_cast, {:ping, 99}},
  {:$gen_cast, {:ping, 100}},
  {:EXIT, #PID<0.274.0>, :shutdown}
]
iex(3)> 12:22:25.636 [info] Foo processed ping-22
12:22:25.636 [info] Foo processing ping-23 ...
12:22:26.637 [info] Foo processed ping-23
12:22:26.637 [info] Foo processing ping-24 ...
12:22:27.638 [info] Foo processed ping-24
12:22:27.638 [info] Foo processing ping-25 ...
12:22:28.639 [info] Foo processed ping-25
12:22:28.639 [info] Foo processing ping-26 ...
12:22:28.721 [info] App stopped
<@graceful-shutdown-demo>-< master>-<±>->
```

▲ v7: reproduce the problem locally

```
iex(1)> 15:09:45.418 request_id=FjKCZGer7yjfov8AAAAC [info] POST /sum
15:09:45.432 request_id=FjKCZGer7yjfov8AAAAC [debug] Processing with CalcWeb.CalcController.sum/2
  Parameters: %{"numbers" => [1, 2, 3, 4]}
  Pipelines: [:api]
15:09:45.434 request_id=FjKCZGer7yjfov8AAAAC [info] Calculating sum of [1, 2, 3, 4]
15:09:45.535 request_id=FjKCZGer7yjfov8AAAAC [info] Sum is 10
15:09:45.535 request_id=FjKCZGer7yjfov8AAAAC [info] Logging operation ...
15:09:45.548 request_id=FjKCZGer7yjfov8AAAAC [info] Sent 200 in 129ms
System.stop()
15:09:46.600 [info] App stopping ...
:ok
iex(2)> 15:09:46.601 [info] App stopped
<@graceful-shutdown-demo>-<./ master>-> █
```

No “Operation logged”

```
<@graceful-shutdown-demo>-<./ master>-> make local-test
curl -X POST http://localhost:4000/sum \
      -H 'Content-Type: application/json' \
      -d '{"numbers": [1, 2, 3, 4]}'
{"result":10}█
<@graceful-shutdown-demo>-<./ master>-> █
```

▲ v7: clean up supervision tree

```
def start(_type, _args) do
  Logger.info("App starting ... ")

  children = [
    {Finch, name: CalcFinch},
    CalcWeb.Telemetry,
    CalcWeb.Endpoint
  ]
```



```
def start(_type, _args) do
  Logger.info("App starting ... ")

  children = [
    Calc.Log,
    CalcWeb.Telemetry,
    CalcWeb.Endpoint
  ]
```

▲ v7: make new Log a Supervisor

```
defmodule Calc.Log do
  use Supervisor
  require Logger

  def start_link(args) do
    Supervisor.start_link(__MODULE__, args, name: __MODULE__)
  end

  @impl true
  def init(_args) do
    children = [
      {Finch, name: __MODULE__.Finch},
      {Task.Supervisor, name: __MODULE__.Task.Supervisor}
    ]

    Supervisor.init(children, strategy: :one_for_one)
  end
end
```

▲ v7: use Task.Supervisor

```
def publish_async(op, numbers, result) do
  Logger.info("Logging operation... ")
  numbers = length(numbers)

  Task.start(fn →
    publish(op, numbers, result)
  end)
end
```



```
def publish_async(op, numbers, result) do
  Logger.info("Logging operation... ")
  numbers = length(numbers)

  Task.Supervisor.start_child(
    __MODULE__.Task.Supervisor,
    fn →
      Process.flag(:trap_exit, true)
      publish(op, numbers, result)
    end,
    shutdown: 7_000
  )
end
```

▲ v7: ok locally

```
15:57:18.464 request_id=FjKE_K2tCbjlpLgAAAEi [info] Calculating sum of [1, 2, 3, 4]
15:57:18.564 request_id=FjKE_K2tCbjlpLgAAAEi [info] Sum is 10
15:57:18.564 request_id=FjKE_K2tCbjlpLgAAAEi [info] Logging operation ...
15:57:18.569 request_id=FjKE_K2tCbjlpLgAAAEi [info] Sent 200 in 118ms
System.stop()
:ok
15:57:19.813 [info] App stopping...
iex(2)> 15:57:23.591 [info] Operation logged
15:57:23.591 [info] App stopped
<@graceful-shutdown-demo>-<[ master>-<±>->
```

▲ v7: ok when deploying

```
running (2m02.0s), 000/500 VUs, 30781 complete and 0 interrupted iterations
default ✓ [=====] 500 VUs 2m0s

✓ HTTP/2
✓ 200 OK

checks.....: 100.00% ✓ 61562 x 0
data_received.....: 110 MB 903 kB/s
data_sent.....: 21 MB 172 kB/s
http_req_blocked.....: avg=1.85s min=468ns med=1.88s max=1.91s p(90)=1.89s p(95)=1.89s
http_req_connecting.....: avg=1.43ms min=0s med=491.5µs max=79.74ms p(90)=3.46ms p(95)=5.63ms
http_req_duration.....: avg=111.56ms min=102.13ms med=106.76ms max=404.97ms p(90)=120.24ms p(95)=130.73ms
http_req_receiving.....: avg=50.38µs min=13.99µs med=45.77µs max=8.29ms p(90)=70.93µs p(95)=80.77µs
http_req_sending.....: avg=193.94µs min=60.57µs med=172.86µs max=8.35ms p(90)=253.22µs p(95)=294.5µs
http_req_tls_handshaking.....: avg=1.84s min=0s med=1.88s max=1.91s p(90)=1.89s p(95)=1.89s
http_req_waiting.....: avg=111.32ms min=101.92ms med=106.52ms max=404.84ms p(90)=119.97ms p(95)=130.47ms
http_reqs.....: 30781 252.226708/s
iteration_duration.....: avg=1.96s min=508.26ms med=1.99s max=2.2s p(90)=2s p(95)=2.01s
iterations.....: 30781 252.226708/s
vus.....: 7 min=7 max=500
vus_max.....: 500 min=500 max=500
```

Connection to 134.209.239.69 closed.

```
<@graceful-shutdown-demo>-[ master>-<±>->
```

```
Every 3.0s: make influxdb-count
```

```
pawel-work-mac.local: Mon Sep 7 22:29:37 2020
```

```
curl -Gs 'http://localhost:8086/query?db=calc' \
--data-urlencode 'q=select count(result) from operation' \
| jq '.results[0].series[0].values[0][1]'
```

```
30781
```

▲ v8: send calcs to InfluxDB in batches of 500 or every 30s

```
defmodule Calc.Log do
  use Supervisor
  require Logger

  def start_link(args) do
    Supervisor.start_link(__MODULE__, args, name: __MODULE__)
  end

  @impl true
  def init(_args) do
    children = [
      {Finch, name: __MODULE__.Finch},
      __MODULE__.Server
    ]

    Supervisor.init(children, strategy: :one_for_one)
  end

  def publish_async(op, numbers, result) do
    __MODULE__.Server.publish(op, numbers, result)
  end
end
```

▲ v8: buffering server

```
defmodule Calc.Log.Server do
  use GenServer
  require Logger

  @publish_size 500
  @publish_interval 30_000

  def start_link(_opts) do
    GenServer.start_link(__MODULE__, :ok, name: __MODULE__)
  end

  def publish(op, numbers, result) do
    timestamp = DateTime.utc_now() |> DateTime.to_unix(:nanosecond)
    request_id = Logger.metadata()[:request_id]
    numbers = length(numbers)
    log = "operation,name=#{op},id=#{request_id} numbers=#{numbers},result=#{result} #{timestamp}"
    GenServer.cast(__MODULE__, {:publish, log})
  end

  @impl true
  def init(:ok) do
    {:ok, %{logs: [], timer: schedule_publishing()}}
  end
```

▲ v8: buffering server

```
@impl true
def handle_cast({:publish, log}, %{logs: {size, logs}} = state)
  when size + 1 < @publish_size do
    {:noreply, %{state | logs: {size + 1, [log | logs]}}}
  end

def handle_cast({:publish, log}, %{logs: {size, logs}, timer: timer}) do
  do_publish(size + 1, [log | logs])
  Process.cancel_timer(timer)
  {:noreply, %{logs: {0, []}, timer: schedule_publishing()}}
end

@impl true
def handle_info(:publish, %{logs: {size, logs}}) do
  do_publish(size, logs)
  {:noreply, %{logs: {0, []}, timer: schedule_publishing()}}
end
```


▲ v9: reproduce the problem locally

```
iex(1)> 00:31:18.409 request_id=FjKhCSpwFxDlpLgAAAAB [info] POST /sum
00:31:18.424 request_id=FjKhCSpwFxDlpLgAAAAB [debug] Processing with CalcWeb.CalcController.sum/2
  Parameters: %{"numbers" => [1, 2, 3, 4]}
  Pipelines: [:api]
00:31:18.425 request_id=FjKhCSpwFxDlpLgAAAAB [info] Calculating sum of [1, 2, 3, 4]
00:31:18.526 request_id=FjKhCSpwFxDlpLgAAAAB [info] Sum is 10
00:31:18.535 request_id=FjKhCSpwFxDlpLgAAAAB [info] Sent 200 in 125ms
System.stop()
:ok
00:31:20.158 [info] App stopping ...
iex(2)> 00:31:20.159 [info] App stopped
<@graceful-shutdown-demo>-<./ master>->
```

No “Operations logged”

▲ v9: just trap exits and send last batch on terminate

```
@impl true
def init(:ok) do
  Process.flag(:trap_exit, true)
  {:ok, %{logs: {0, []}, timer: schedule_publishing()}}
end
```

```
@impl true
def terminate(_reason, %{logs: {size, logs}}) do
  do_publish(size, logs)
end
```

▲ v9: ok locally

```
iex(1)> 00:36:19.394 request_id=FjKhTz6S3UDlpLgAAAAF [info] POST /sum
00:36:19.409 request_id=FjKhTz6S3UDlpLgAAAAF [debug] Processing with CalcWeb.CalcController.sum/2
  Parameters: %{"numbers" => [1, 2, 3, 4]}
  Pipelines: [:api]
00:36:19.411 request_id=FjKhTz6S3UDlpLgAAAAF [info] Calculating sum of [1, 2, 3, 4]
00:36:19.512 request_id=FjKhTz6S3UDlpLgAAAAF [info] Sum is 10
00:36:19.518 request_id=FjKhTz6S3UDlpLgAAAAF [info] Sent 200 in 123ms
System.stop()
:ok
00:36:20.662 [info] App stopping ...
iex(2)> 00:36:20.663 [info] Logging 1 operations ...
00:36:20.696 [info] Operations logged
00:36:20.696 [info] App stopped
<@graceful-shutdown-demo>-< master>-<±>->
```

▲ v9: ok when deploying

```
running (2m02.6s), 000/500 VUs, 30555 complete and 0 interrupted iterations
default ✓ [=====] 500 VUs 2m0s

✓ HTTP/2
✓ 200 OK

checks.....: 100.00% ✓ 61110 x 0
data_received.....: 109 MB 892 kB/s
data_sent.....: 21 MB 170 kB/s
http_req_blocked.....: avg=1.86s min=361ns med=1.88s max=2.28s p(90)=1.89s p(95)=1.89s
http_req_connecting.....: avg=1.22ms min=0s med=448.49µs max=76.35ms p(90)=1.79ms p(95)=4.65ms
http_req_duration.....: avg=113.8ms min=102.28ms med=106.61ms max=1.13s p(90)=126.01ms p(95)=141.84ms
http_req_receiving.....: avg=47.4µs min=13.86µs med=43.38µs max=3.61ms p(90)=66.51µs p(95)=76.01µs
http_req_sending.....: avg=183.39µs min=45.97µs med=165.4µs max=16.28ms p(90)=240.06µs p(95)=274.88µs
http_req_tls_handshaking ..: avg=1.86s min=0s med=1.88s max=2.28s p(90)=1.89s p(95)=1.89s
http_req_waiting.....: avg=113.77ms min=102.05ms med=106.37ms max=1.13s p(90)=125.77ms p(95)=141.59ms
http_reqs.....: 30555 49.152351/s
iteration_duration.....: avg=1.77s min=441.17ms med=1.99s max=3.03s p(90)=2.01s p(95)=2.02s
iterations.....: 30555 249.152351/s
vus.....: 8 min=8 max=500
vus_max.....: 500 min=500 max=500

Connection to 134.209.239.69 closed.
```

```
<@graceful-shutdown-demo>->/ master>-> □
```

```
Every 3.0s: make influxdb-count
```

```
pawel-work-mac.local: Tue Sep 8 00:45:49 2020
```

```
curl -Gs 'http://localhost:8086/query?db=calc' \
--data-urlencode 'q=select count(result) from operation' \
| jq '.results[0].series[0].values[0][1]'
```

```
30555
```



Key points

1. Test your graceful shutdown (continuously) in Staging and in Prod
2. Understand how you run and deploy services
3. It's easy to implement graceful shutdown in Elixir and OTP

Thank you

https://github.com/pawel-szafran/elixir_graceful_shutdown_demo

<https://telnyx.com/company/careers>

