

Willst du.
Kannst du.

karriere.at

ÖSTERREICH'S GRÖSSTES JOBPORTAL

About me

- karriere.at
 - Lead DevOps Engineer
 - Journey
 - From Snowflake to Ansible
 - From OnPrem to Cloud
 - Making Laravel Services Cloud Ready
 - Moving from DevOps to Platform Engineering





karriere.at als Arbeitgeber

karriere.at

karriere.at Group: Unsere Vision

Alle arbeitsfähigen
Personen in Österreich

Wir stellen sicher, dass Kandidat*innen
passende Stellenangebote erhalten
(pull/push) und sich möglichst schnell
über unsere Systeme bewerben

**Jede*r findet den passenden Job
in Angeboten der karriere.at Gruppe**

Je nach Marktsegment:
karriere.at, jobs.at, hokify

Purpose



Wir verbinden **Unternehmen** und **Talente**.

*Mehr als 250
Mitarbeiter*innen*

davon ca. 70 im
Technology Bereich





2 Standorte

Linz und Wien



DOWN THE RABBIT HOLE:

Lessons learned shifting Laravel services to Kubernetes

Cloud Native Linz, 18.03.2025

Our Cloud Native Journey

- Started around 2 years ago
- Mainly based on GCP, Kubernetes, Flux and Gitlab
- PHP / Laravel Stack
 - Containerized Deployments can be challenging
 - Many mechanisms to make it stable and secure
 - Many services to be managed
- Transformation from monolith to microservices

Running Services the Cloud Native Way

- Build ONE container image
- Deploy the image to Kubernetes using a deployment
- Expose it using a service



PHP is different

PHP

- Request based language
- Interpreted language
- No binary, a lot of separate files
- Eventually a frontend in form of html, js and css files

▼ LARAVEL-SERVICE

- > app
- > bootstrap
- > config
- > database
- > public
- > resources
- > routes
- > specification
- > storage
- > vendor
- ≡ artisan
- { } composer.json
- { } composer.lock

Deployment

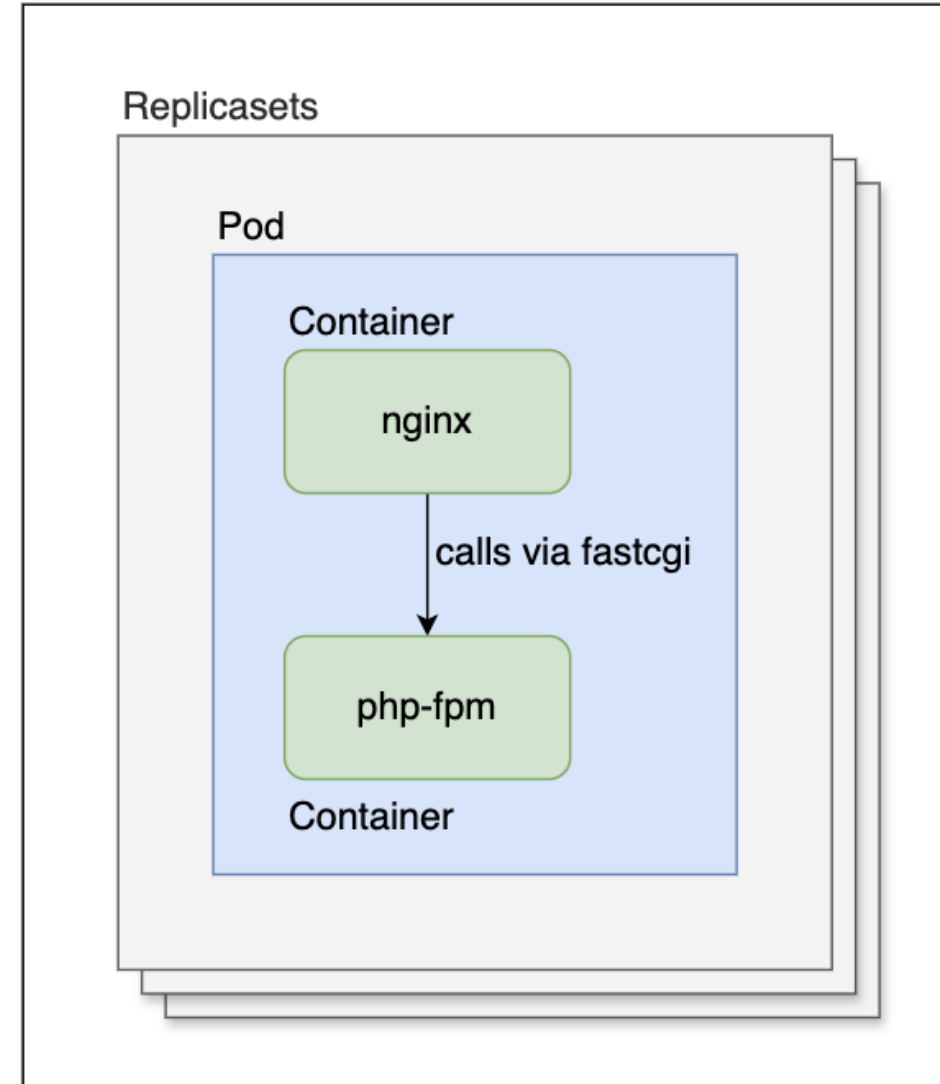
- Apache + Mod PHP
- Nginx + PHP-FPM ← we are here
- PHP Application Servers
 - FrankenPHP
 - Swoole

PHP in a Cloud Native World

Approach 1 – Two Containers

- One Process per container
 - Nginx container
 - PHP-FPM container
- Issues
 - All files of the PHP application need to be added to both containers
 - Monitoring of the PHP container is not easy
 - Scaling and management of two containers

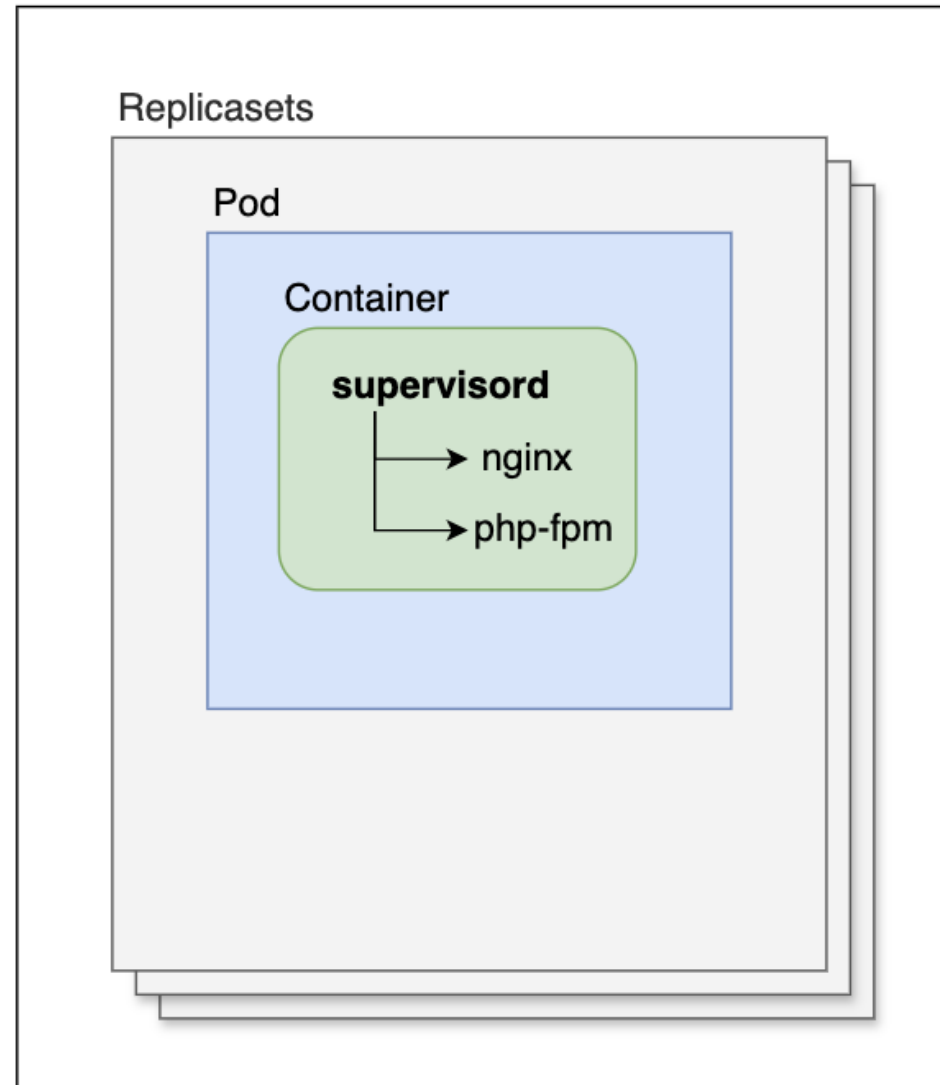
Deployment



Approach 2 – One Container

- Most popular approach in the community
- Using supervisord as process manager
- All files of the PHP app in one container
- Monitor Nginx and PHP-FPM combined using http health checks
- Only one container to manage and maintain

Deployment



So, are we finished now?!



Not quite...

```
<?php

namespace App\Http\Controllers\Actions;

use App\Jobs\Application\SendCopyToUser;
use ...

class SendCopyToUserController extends Controller
{
    public function __invoke(string $applicationUuid): Response
    {
        SendCopyToUser::dispatch(...);
        return response()->noContent();
    }
}
```

← Async Task

Async Tasks in Laravel

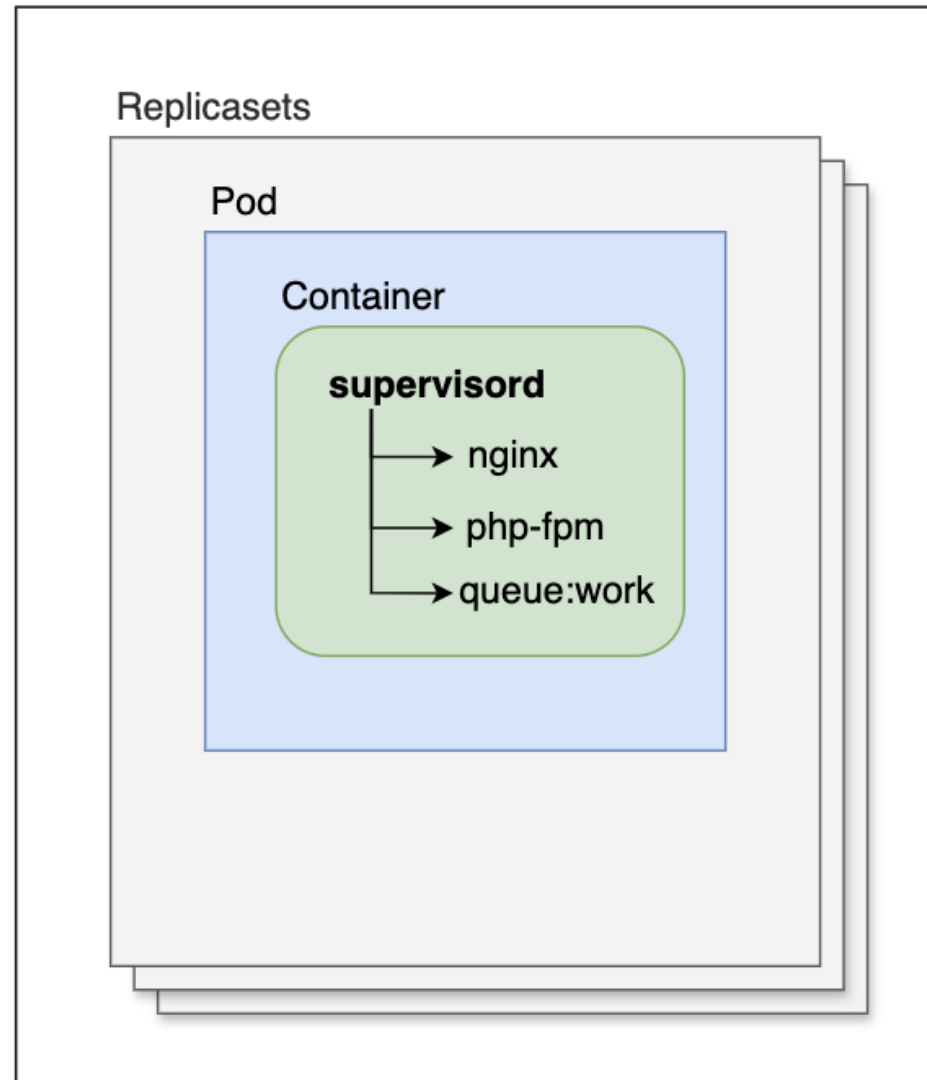
- Laravel uses a queue worker
 - Separate process
 - Uses e.g. Redis as queue for async tasks
- How to deploy this in a cloud native way?

```
php artisan queue:work
```

Approach 1 – supervisord

- Simply add another supervisord process?
- Going rouge? Dying? - Unnoticed!
- This approach can impact performance of the webapp
- No supervisord for PHP application servers like FrankenPHP

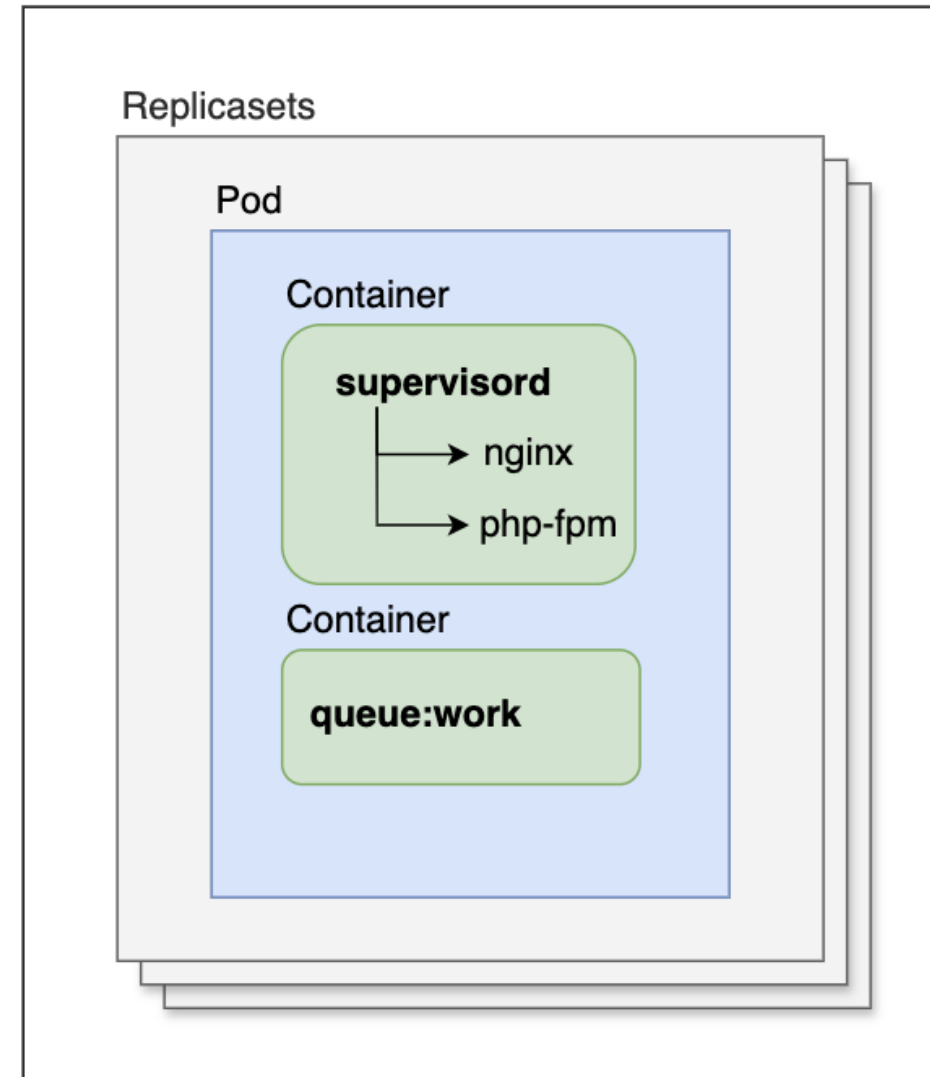
Deployment



Approach 2 – Sidecar

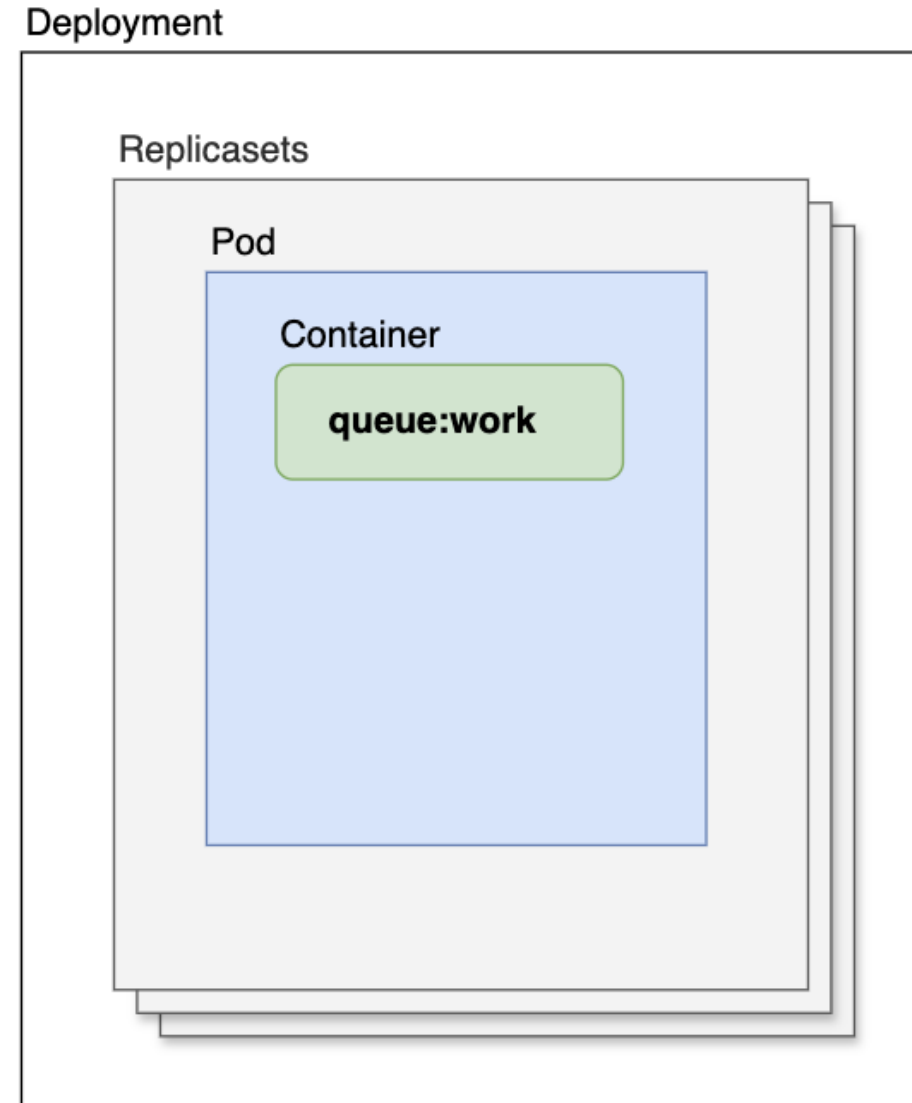
- Add a sidecar container to the existing deployment in Kubernetes
- Always scaled with webapp
- Scaling requirements for queue worker are different than for webservice

Deployment



Approach 3 – Separate Deployment

- Components need to be split based on failure and scaling domains
- Use a separate Kubernetes deployment for the queue worker to allow for separate scaling



So, are we finished now?!

Developer 12:49
I want to run tasks every day at 1:00 🚀

DevOps 19:00
Cool there is a ready to use thing in k8s -> CronJob

Developer 19:01
I implemented that with a CronJob but my Tasks fail shortly after scheduler is executed 😞

DevOps 19:01
Ahm? 🤔

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: Laravel Scheduler
spec:
  schedule: "* * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: hello
              image: laravel-service
              imagePullPolicy: IfNotPresent
              command:
                - php
                - artisan
                - schedule:run
              restartPolicy: OnFailure
```


Laravel Scheduler

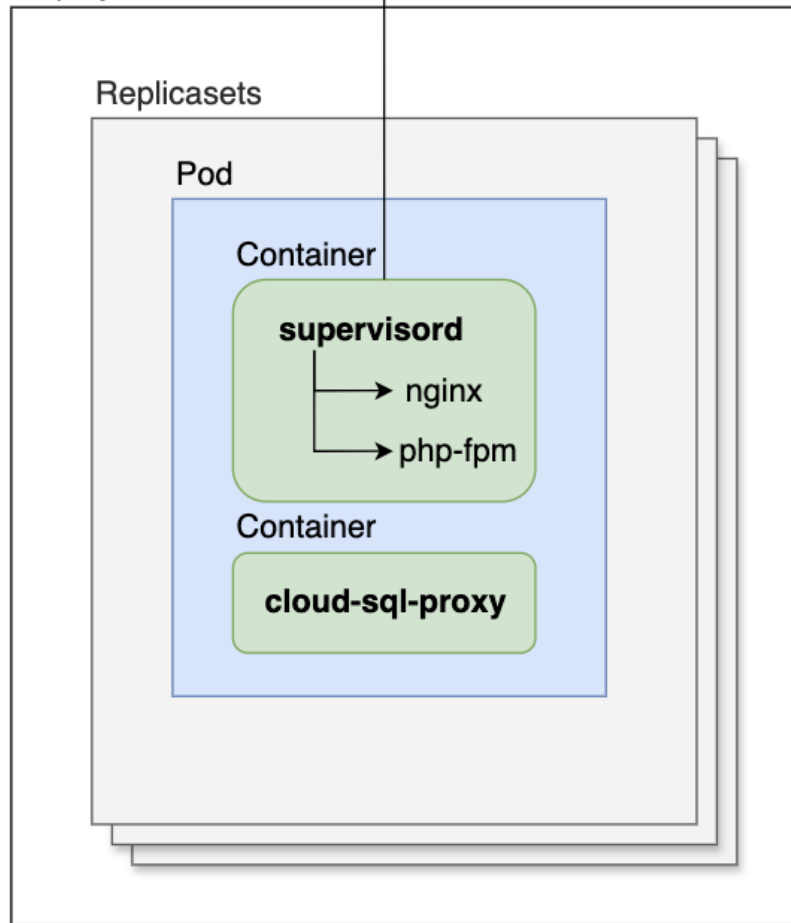
- Laravel scheduler can execute tasks in background and the main process exists within seconds after start
- Laravel scheduler can also be executed as long running process
- We already did that with the queue:worker
- Only difference is scaling – Only 1 instance

```
php artisan schedule:work
```

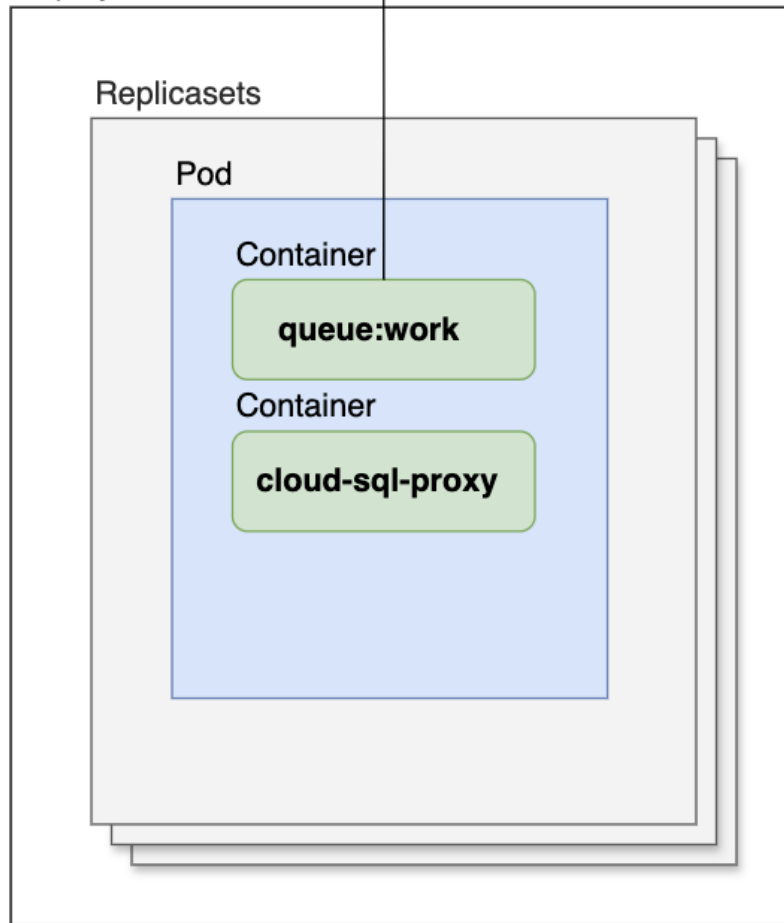
```
$schedule->command(SendJobReminderEmailCommand::class)  
->runInBackground()  
->dailyAt('06:00')  
->onOneServer();
```

ConfigMap

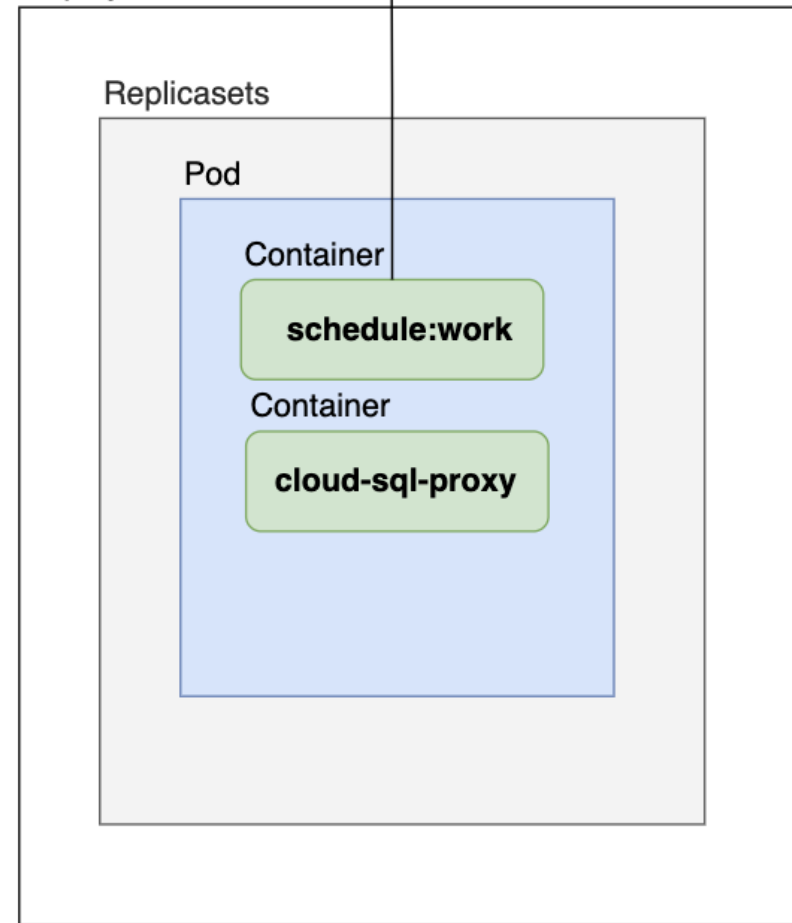
Deployment



Deployment



Deployment



Outlook FrankenPHP/Swoole

- No supervisord needed
- Performance improvement
- Laravel Octane needed
- Change from request based to long running process
- Issues with Open Telemetry auto instrumentation in FrankenPHP
- Does not solve scheduler and queue worker

We are hiring



smart bewerben ▶ EINBLICKE VIDEOS

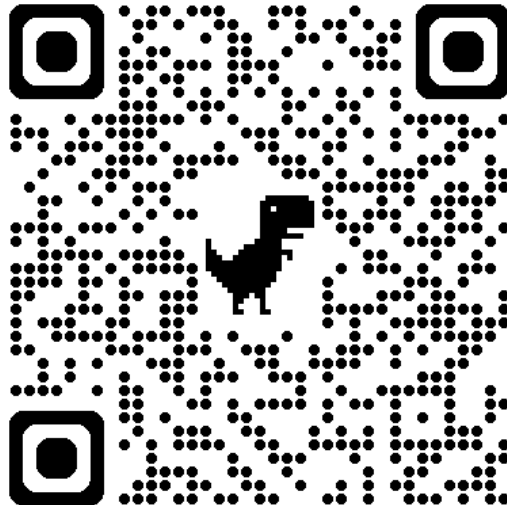
DevOps/Cloud Engineer*

karriere.at

vor 6 Tagen veröffentlicht

Linz, Wien Vollzeit Homeoffice

3.843 € – 5.000 € monatlich



Q&A- Runde

Feel free to ask questions or
reach out

sebastian.huber@karriere.at

karriere.at