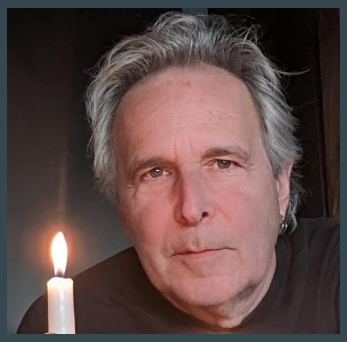
Almost Serverless:



Barcelona AWS Meetup
2023-11-30

Chris Shenton
CTO V! Studios

I've been working on AWS since 2013 when we won a NASA contract to migrate hundreds of applications into the cloud, and build new cloud-native services.
I've done robotics and AI, networking, security, penetration testing, built Internet Service Providers, and like designing fault-tolerant architectures.
Serverless is my favorite tech, and I'm a Python Guy.

Almost Serverless:

•••

Barcelona AWS Meetup
2023-11-30

Chris Shenton
CTO V! Studios

V! Studios is a small firm outside Washington DC; our clients include government (NASA), commercial, and non-profit organizations.
Part of our team does 3D rendering and visualization, another part provides strategic communications, and my team builds cloud-based applications in AWS

Almost Serverless:
Scale to Zero

•••

Barcelona AWS Meetup
2023-11-30

Chris Shenton
CTO V! Studios

This is a Serverless Meetup, but we've got apps that don't fit in Lambda or DynamoDB.
I wanted to see how close I could get to Serverless

Almost Serverless:
Scale to Zero
with
App Runner and Aurora DB

•••

Barcelona AWS Meetup
2023-11-30

Chris Shenton
CTO V! Studios

We'll walk through managed Docker and SQL services.

Almost Serverless:
Scale to Zero
with
App Runner and Aurora DB
running
Wagtail CMS on Django

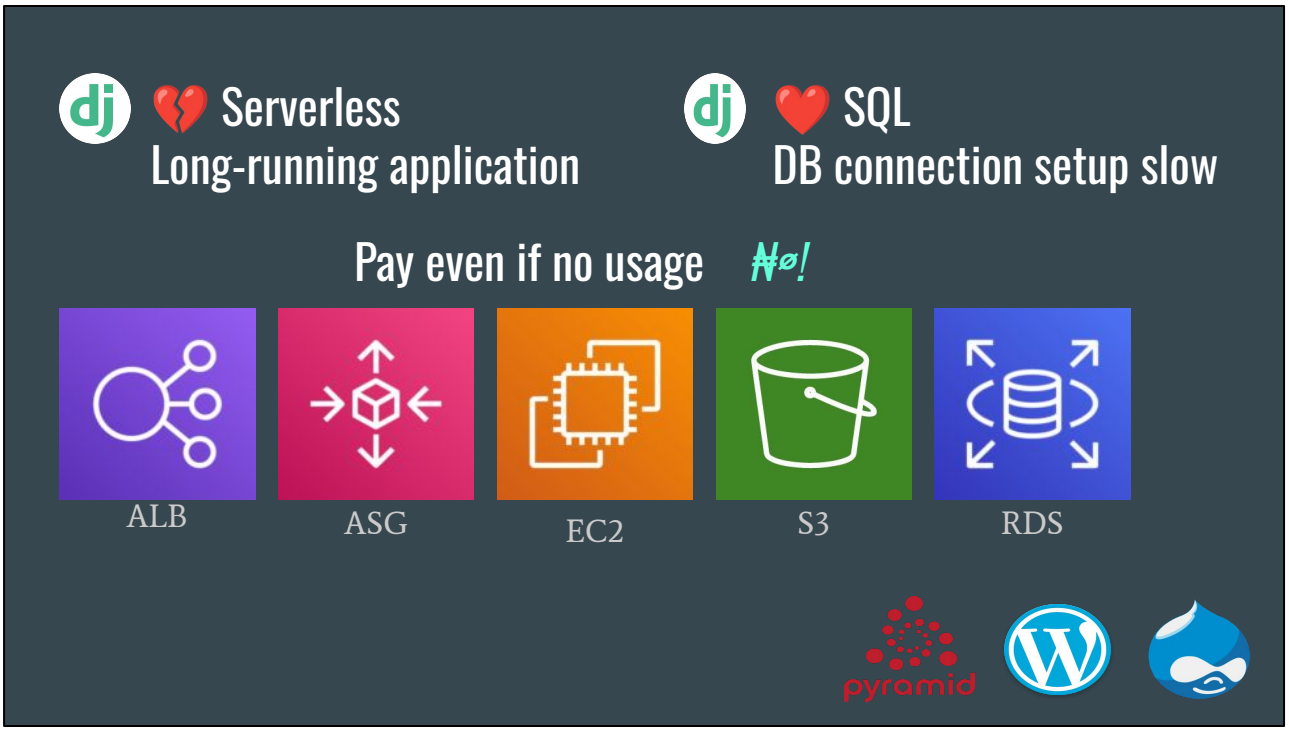
•••

Barcelona AWS Meetup
2023-11-30

Chris Shenton
CTO V! Studios

Our target apps run on Wagtail CMS, which runs on Django, the popular Python web framework.

Compute, Database, Storage, Messaging, Orchestration: all scale to zero, no cost when unused.
Open Search Serverless has a minimum of about 700€/month which doesn't meet my definition of "serverless".
Most icons from https://acloudxpert.com/aws-icon-set/



Django, Wordpress, Drupal use SQL DB.
Pyramid can use SQL or NoSQL DBs (like DynamoDB)
The stylized N is the Nigeria Naira currency symbol: ₦

Evolution of our Code Deployment

Complex.
Slow to start:
30 minutes.

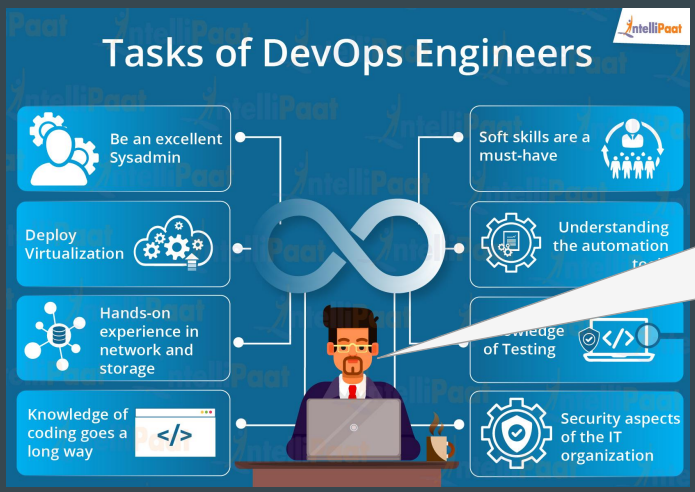
Scripts install code, configure machine.

Reliable.
Quick to start:
3 minutes.

Docker image is the deployment package.

We first used Ansible to install code and configure EC2,
then switched to building Docker images and shipping those to EC2.
But we don't really want to manage EC2: security upgrades, AMI deprecations, etc.

But I just want to build apps -- make DevOps go away!



If only there was a
managed service
for Docker containers!

Graphic from https://intellipaat.com/blog/wp-content/uploads/2017/11/DevOps-01.jpg

Hold My Beer!

The 17 Ways to Run Containers on AWS

17 More Ways to Run Containers on AWS

Corey Quinn

Chief Cloud Economist at The Duckbill Group
Screaming in the Cloud / Last Week in AWS

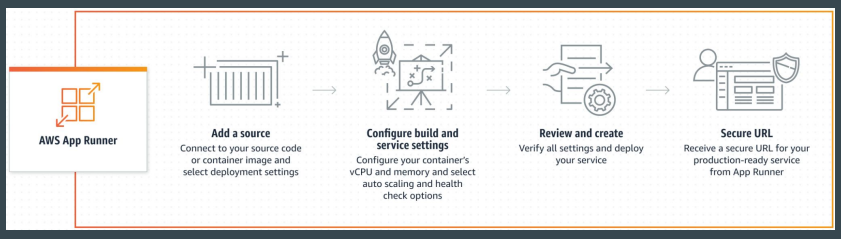
https://www.lastweekinaws.com/blog/the-17-ways-to-run-containers-on-aws/
https://www.lastweekinaws.com/blog/17-more-ways-to-run-containers-on-aws/

Comparison of docker deployment environments (YMMV)

	EC2 ASG ALB	K8S	EKS	ECS	ECS on Fargate	App Runner
Complexity	You do everything	You do everything, plus managing K8S itself	AWS manages K8S	You manage the underlying EC2.	AWS manages EC2. You define ECS: VPC, subnets, routing, IGW, SGs, ALB, Scaling Policy, Task (container image(s), cpu, …)	AWS manages everything.
Benefits	Many AWS integrations	Portable across vendors	You don't have to run K8S yourself.	Many AWS Integrations. CloudFormation support.	AWS runs the underlying ECS/EC2 infrastructure.. Flexible, can run multi-container apps	Provide image from ECR/GitHub, App Runner runs and scales it automatically. VPC access. Redeploys upon new image. No charges for load balancing. **Scale to Zero!**
Limitations	Scale out is rather slow.	Least common denominator, no managed services	AWS only. AWS integrations may be limited	AWS only	Container cannot run in privileged mode. No GPU or EBS volumes.	Single application per App Runner. HTTP only. No priv mode, GPU, EBS, EFS.
Cost	Depends on app infrastructure. Probably lowest since no managed services	Depends on app infrastructure, plus K8S infrastructure.	$72/month/cluster + compute resources	ECS is a free service, just cost of your compute resources.	Task: 1 vCPU @ $29 + 4 GB RAM @ $13 = $42/month	For 24x7, about 60% more than Fargate. If you scale to zero, you could save a lot, like 75%. Best for "small" workloads with long idles.

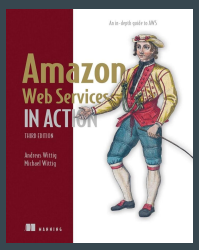
Fargate is about 50% more expensive than bare EC2, but requires much less labor: "In our opinion, Fargate is worth it in most scenarios" -- AWS Web Services in Action 3rd edition.

App Runner



❝ App Runner is a Platform as a Service (PaaS) offering for container workloads. You provide a container image bundling a web application, and App Runner takes care of everything else:

- Runs and monitors containers
- Distributes requests among running containers
- Scales the number of containers based on load
- You pay for memory but not CPU resources during times when a running container does not process any requests. ❞



Andreas Wittig,
Michael Wittig

Automatically deploys runs the new version when you upload a new image.
Active: $0.064/ vCPU-hour + 0.007 / GB-hour = $0.071/hour = 0.070€/hour
Idle: 0.007 / GB-hour = 0.007/hour = 0.006€ (about 10% of Active)
https://aws.amazon.com/apprunner/pricing/

Database: Aurora Serverless v1

❝ Aurora Serverless is an on-demand, auto scaling configuration that automatically adjusts database capacity based on application needs. ... and shuts down during periods of inactivity, saving you money and administration time. ❞
https://aws.amazon.com/rds/aurora/pricing/

❝ Pause after inactivity – Use the optional Scale the capacity to 0 ACUs when cluster is idle setting to scale the database to zero processing capacity while it's inactive. When database traffic resumes, Aurora automatically resumes processing capacity and scales to handle the traffic. ❞

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.modifying.html

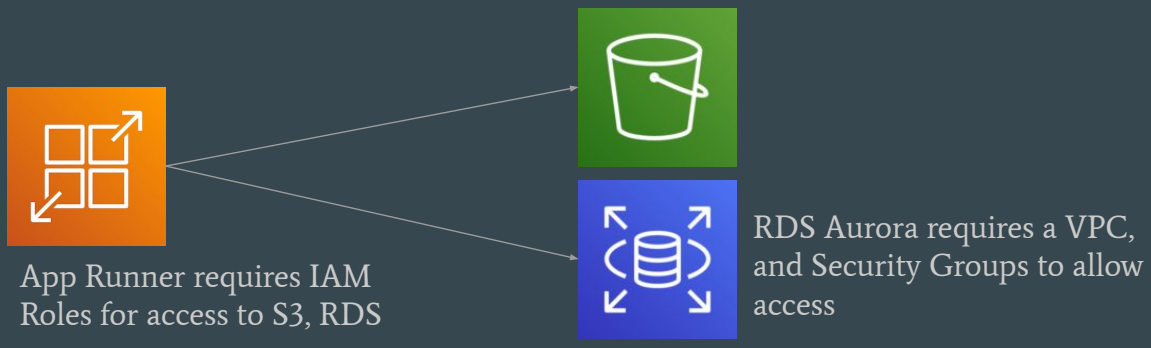
Introduced early 2018
$0.07 / ACU-Hour * Minimum 2 ACU = $0.14 / hour = 0.13€ / hour
https://aws.amazon.com/rds/aurora/pricing/

App Runner: AWS does all the work

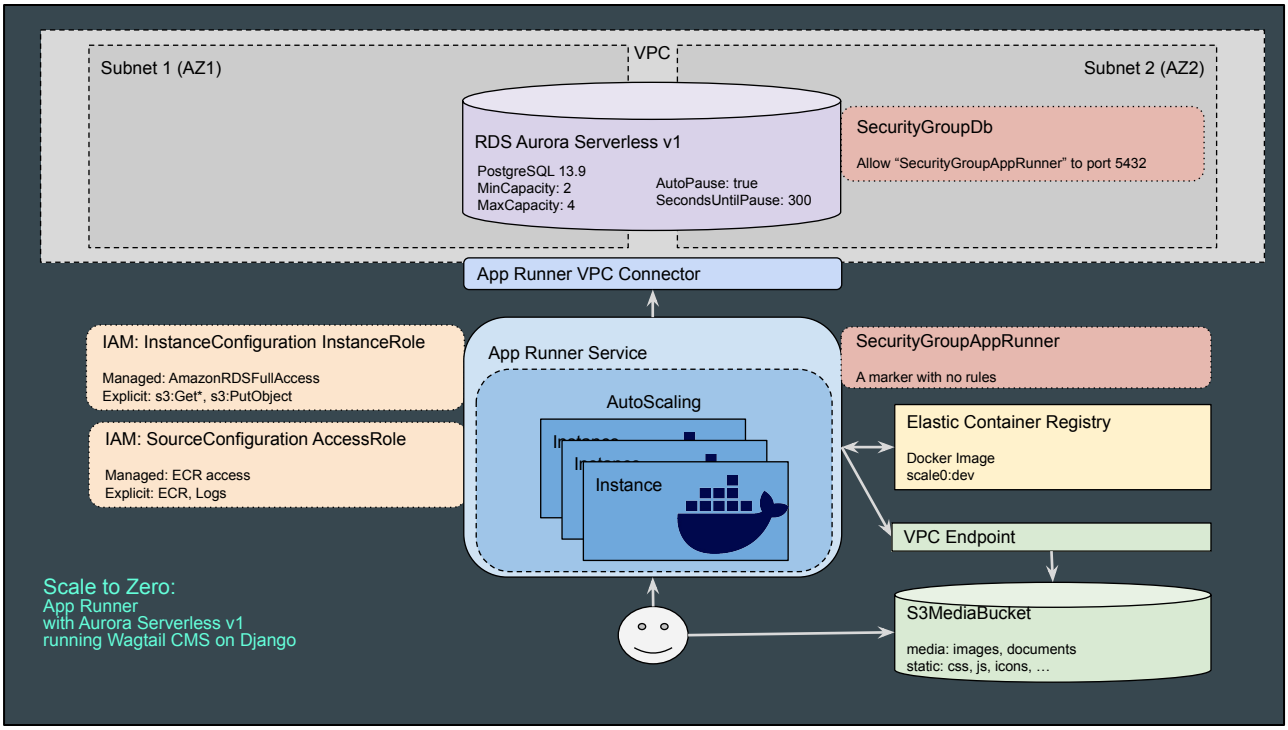
```
aws apprunner create-service \
    --service-name simple \
    --source-configuration '{"ImageRepository": \
      {"ImageIdentifier": "public.ecr.aws/s5r5a1t5/simple:latest", \
       "ImageRepositoryType": "ECR_PUBLIC"}}'
```

Code from Wittig & Wittig “Amazon Web Services in Action, Third Edition.

App Runner: stateless services need external storage

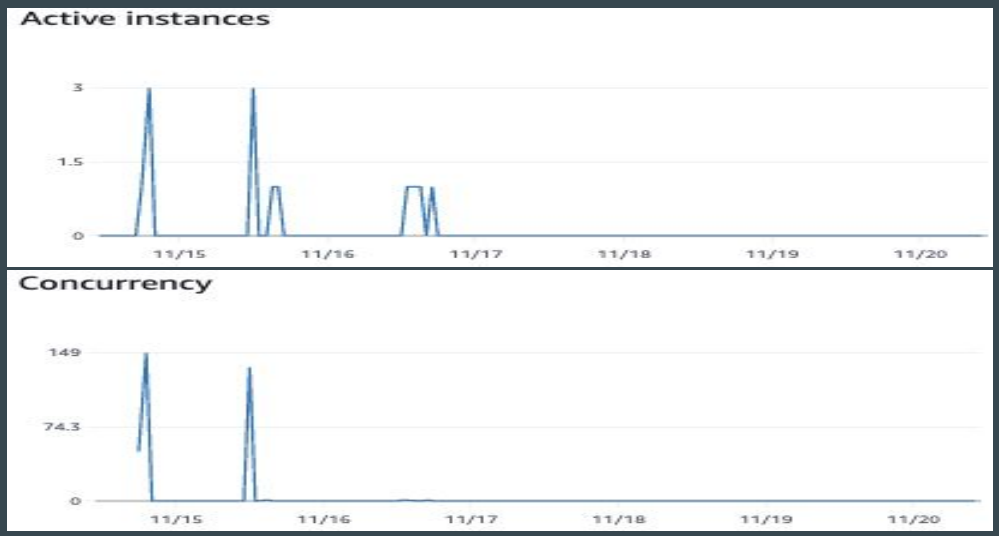


External persistence allows scaled-out and new App Runner containers to get the data.
App Runner got VPC access in February 2022.
https://aws.amazon.com/blogs/aws/new-for-app-runner-vpc-support/

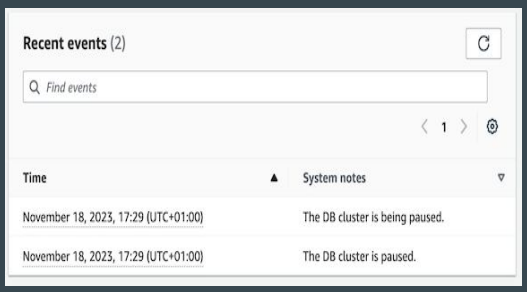


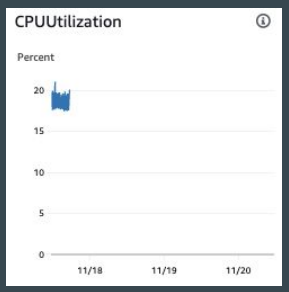
App Runner appears to be in its own VPC, and I had to add a VPC Endpoint for S3 access.
I’m using public subnets so I don’t have to pay $70/month for NATGW;
the VPC and Security Groups keep it safe.
There’s also no ALB, so we save another $20/month.

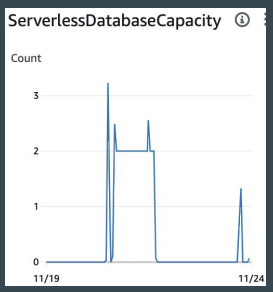
App Runner: scales to zero



Aurora Serverless v1: scales to zero







The database is “paused”

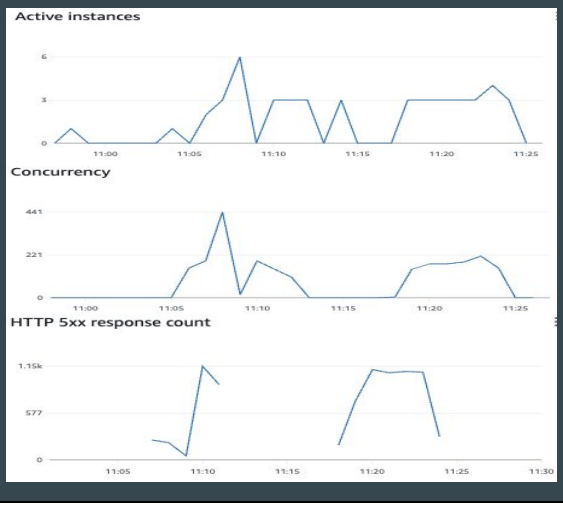
Aurora Serverless v1: DB takes about 30 seconds to wake up

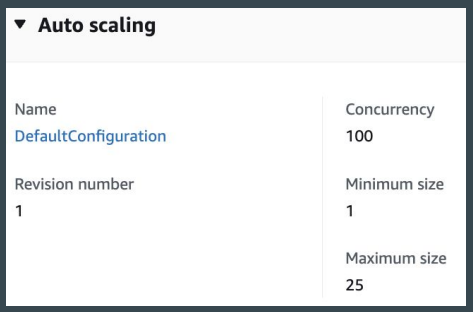
```
% time curl https://ykcgyztfmf.eu-west-3.awsapprunner.com > /dev/null
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 10686  100 10686    0     0    353      0  0:00:30  0:00:30 --:--:--  2756
0.02s user 0.03s system 0% cpu 30.249 total


% time curl https://ykcgyztfmf.eu-west-3.awsapprunner.com > /dev/null
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 10626  100 10626    0     0  48692      0 --:--:-- --:--:-- --:--:-- 49654
0.01s user 0.01s system 8% cpu 0.227 total
```

Wake up, little server, wake up!
First connection wakes it up, in 30 seconds;
second is served quickly.

App Runner: scales on load

```
% hey -z 5m -c 201 https://ykcgyztfmf.eu-west-3.awsapprunner.com
```





HTTP 500 errors on scale out: app logs indicate PostgreSQL doesn't have enough incoming connections, DB hasn't scaled out quickly enough.

Hey load tester: https://github.com/rakyll/hey
Set to scale out when request concurrency above 100.
HTTP 500 errors: in app logs: "FATAL: remaining connection slots are reserved for non replicate superuser connections"

Time for a live demo?

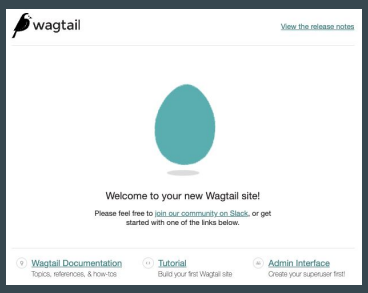
Go to app by URL:
https://ykcgyztfmf.eu-west-3.awsapprunner.com/

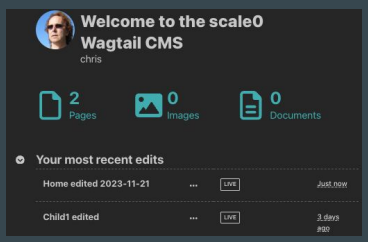
Wait 30 seconds for Aurora to wake up.

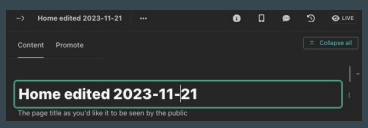
Make a change to the home page.

Load test for 1 minute:

```
hey -z 1 m -c 25 \
https://ykcgyztfmf.eu-west-3.awsapprunner.com/
```







Demo windows:

- Terminal with: hey -z 1m -c 25 https://ykcgyztfmf.eu-west-3.awsapprunner.com/
- App Runner console showing graphs
- Wagtail Admin window

Wrap Up: 👍

App Runner

Easiest way to deploy web-based apps.

Monthly cost scenarios:

- Dev/test: 5€
- Light prod (8 hour/day): 23€
- High volume: 93€

https://aws.amazon.com/apprunner/pricing/

Aurora Serverless v1

Perfect for development and test environments.

Monthly Cost Scenario:

- 2ACU minimum (5 days/week, 8 hour/day): 22€/month

https://aws.amazon.com/rds/aurora/pricing/

While scaled-to-zero isn't free, 30€/month for a developer environment seems reasonable.

Other options for long running apps with SQL DBs

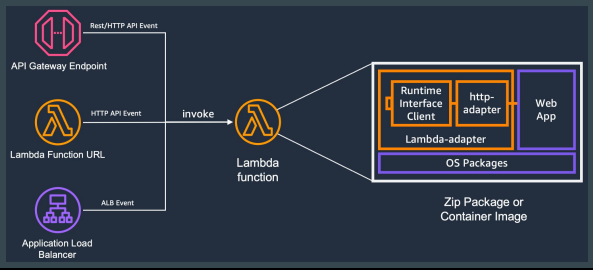
Aurora Serverless v2

❝ instantly scales databases to support hundreds of thousands of transactions per second... customers can lower costs ... due to a lower starting capacity of 0.5 ACUs (compared to 2 ACUs in v1), ... and up to 15x faster scale down in capacity compared to Amazon Aurora Serverless v1.❞
https://aws.amazon.com/rds/aurora/pricing/

(minimum cost is about 40€/month)

Lambda Web Adapter

❝allows developers to build web apps (http api) with familiar frameworks (e.g. Express.js, Next.js, Flask, SpringBoot, ASP.NET and Laravel, anything speaks HTTP 1.1/1.0) and run it on AWS Lambda. The same docker image can run on AWS Lambda, Amazon EC2, AWS Fargate, and local computers.❞
https://github.com/awslabs/aws-lambda-web-adapter



LWA first release September 2022, current version 0.7 released August 2023.
Lambda Web Adapter points out that this makes your app portable across local, EC2, Fargate, and Lambda (or AppRunner).

Resources

Code and presentation:
https://github.com/v-studios/scale-to-zero

Me:
@shentonfreude
@shentonfreude@mastodon.online

Company:
chris@v-studios.com
https://v-studios.com
https://www.linkedin.com/company/v-studios/

Amazon Web Services in Action, third edition
by Andreas Wittig and Michael Wittig
Chapter 18 on ECS, Fargate, App runner
https://cloudonaut.io/review-apprunner-simply-containers-on-aws/

The book is light on App Runner, but has lots of good info on Fargate.