

Fargate

Aurelijus Banelis

VilniusPHP 0x63
2021-02-04





Aurelijus Banelis

Senior Software Engineer

aurelijus.banelis.lt
aurelijus@banelis.lt

PGP 0x320205E7**539B6203**
130D C446 1F1A 2E50 D6E3
3DA8 3202 05E7 539B 6203



home 24





Aurelijus Banelis

Senior Software Engineer

aurelijus.banelis.lt
aurelijus@banelis.lt

PGP 0x320205E7**539B6203**
130D C446 1F1A 2E50 D6E3
3DA8 3202 05E7 539B 6203



home24



A dark, grayscale photograph of a forest scene. The ground is covered in snow, and the trees are mostly bare, with some evergreen branches visible. The lighting is low, creating a somber atmosphere.

Development
with
infrastructure



What

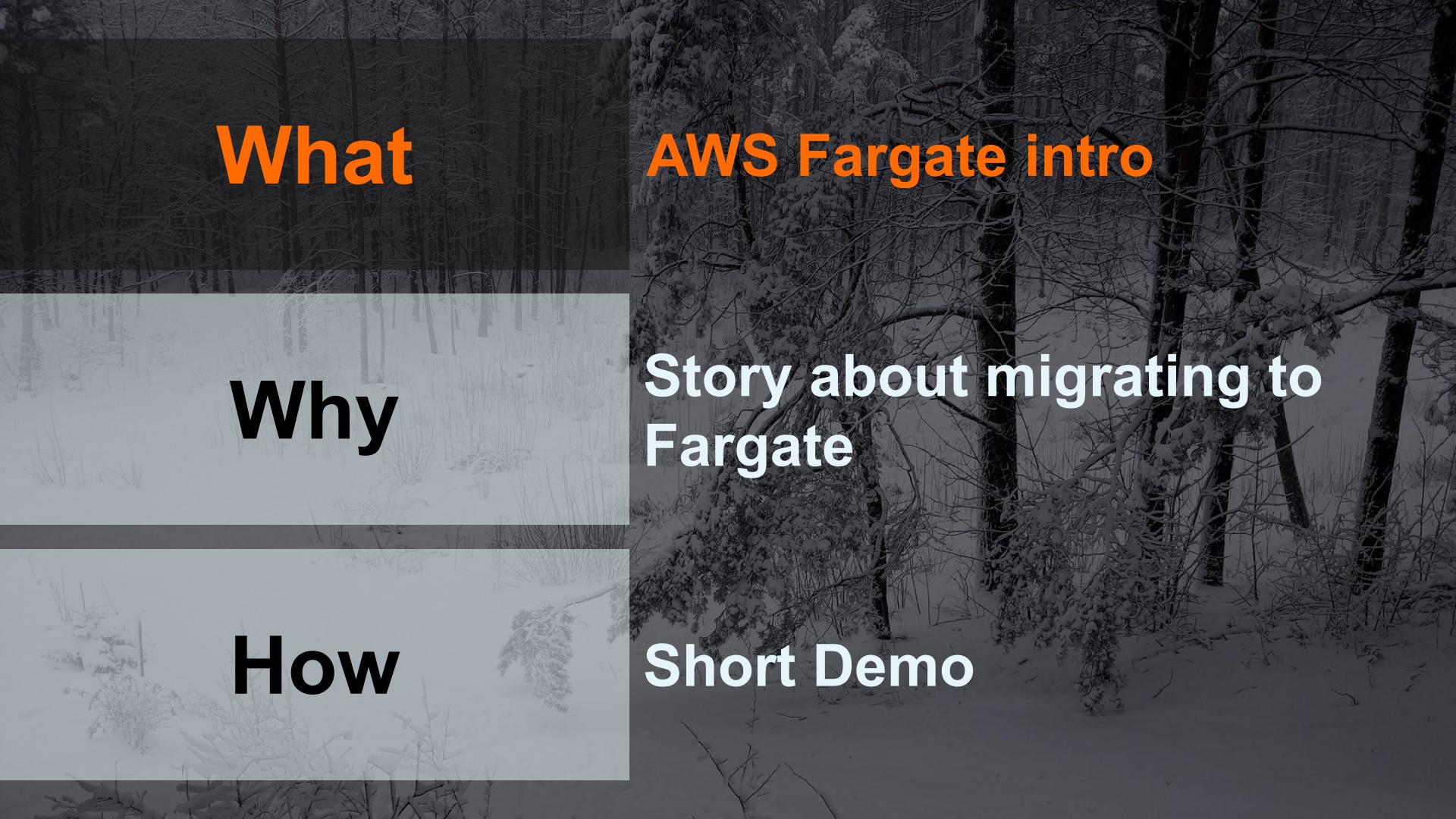
Why

How

AWS Fargate intro

Story about migrating to
Fargate

Short Demo



What

AWS Fargate intro

Why

**Story about migrating to
Fargate**

How

Short Demo

AWS Fargate

Serverless compute for containers

Get started with AWS Fargate

Serverless compute for containers

FEATURED LAUNCH

AWS Proton

Define, manage, and update your infrastructure so your developers can focus on writing great code.

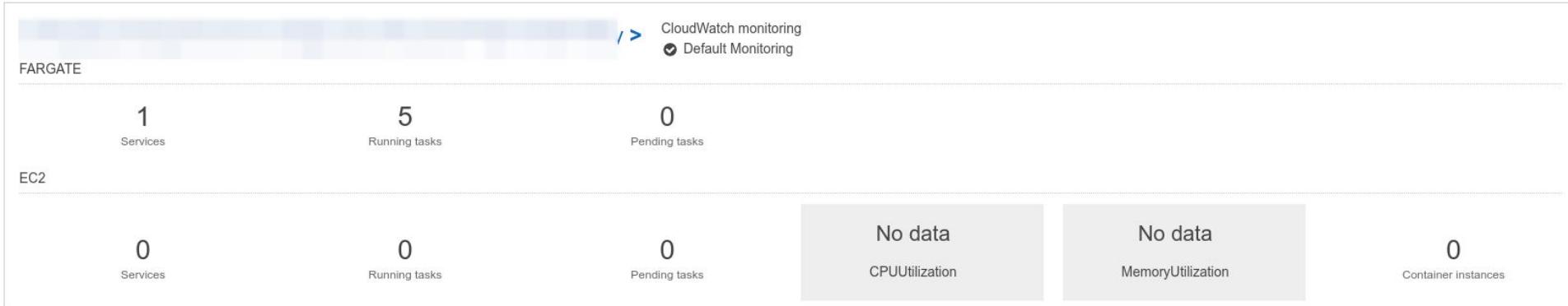
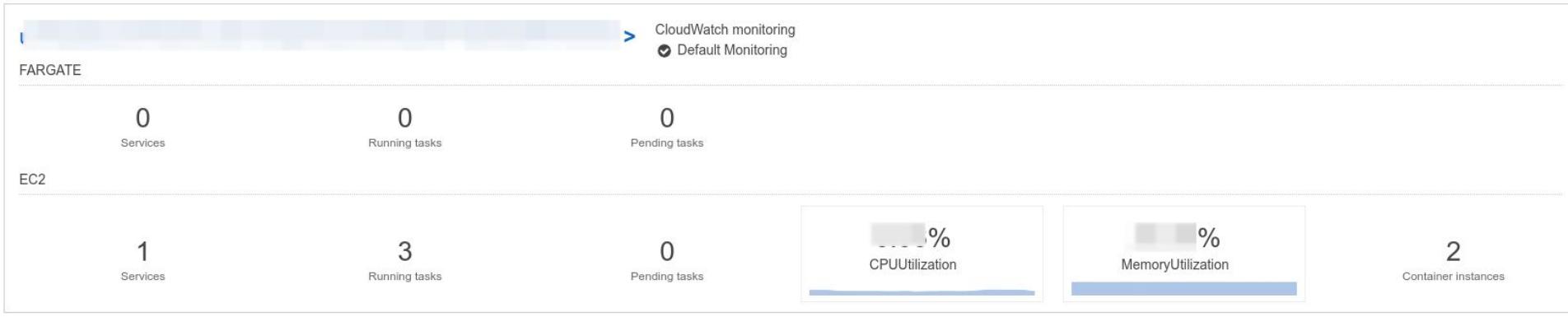
Learn more ➔

AWS Fargate is a serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Fargate makes it easy for you to focus on building your applications. Fargate removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation by design.

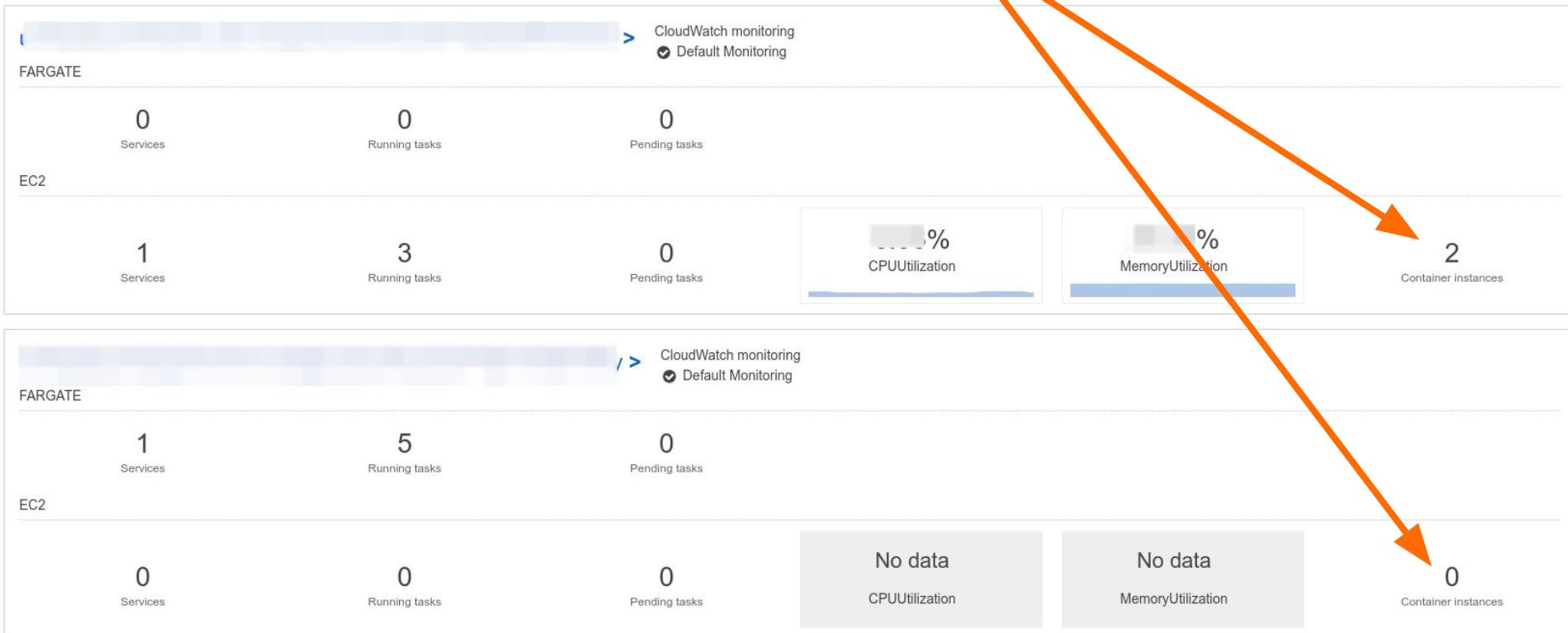
Fargate allocates the right amount of compute, eliminating the need to choose instances and scale cluster capacity. You only pay for the resources required to run your containers, so there is no over-provisioning and paying for additional servers. Fargate runs each task or pod in its own kernel providing the tasks and pods their own isolated compute environment. This enables your application to have workload isolation and improved security by design. This is why customers such as Vanguard, Accenture, Foursquare, and Ancestry have chosen to run their mission critical applications on Fargate.

<https://aws.amazon.com/fargate>

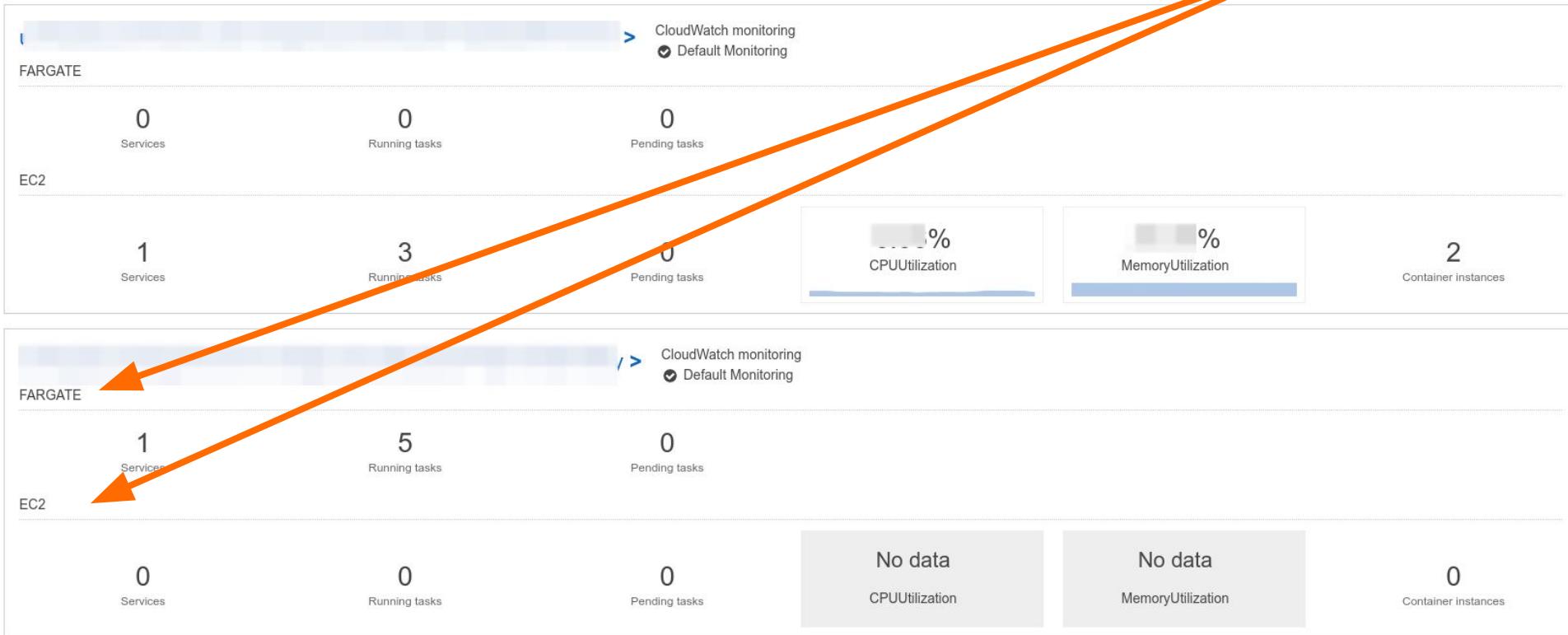
Serverless compute



Serverless compute



Serverless compute







Lambda



Fargate



EC2



docker



Lambda



Fargate



docker



EC2



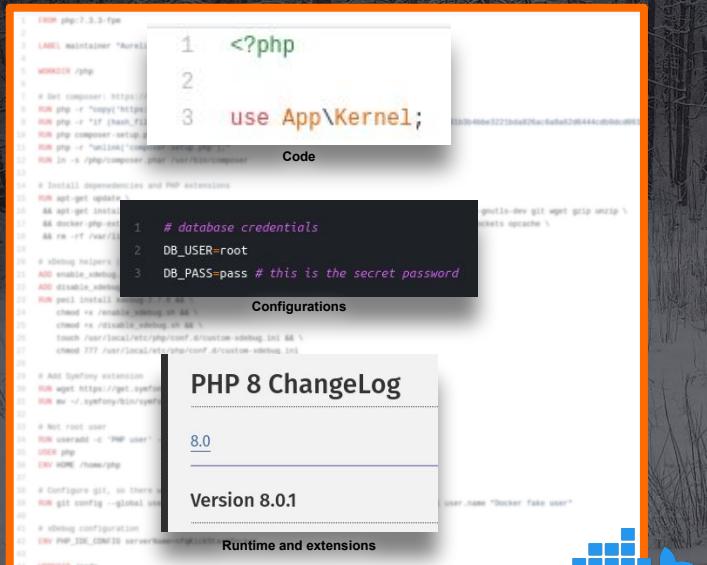
docker

For containers



For containers

For containers



```
FROM php:7.3.3-fpm

LABEL maintainer "Nordi"
WORKDIR /app

RUN composer https://
RUN php -r "if (hash('
RUN php composer-setup.php
RUN curl -s https://getcomposer.org/installer | ph
RUN curl -s https://getcomposer.org/installer | ph

# Install dependencies and PHP extensions
RUN apt-get update
RUN apt-get install -y libxml2-dev libcurl4-openssl-dev libzip-dev libon
RUN docker-php-ext-configure --with-zip --with-on
RUN docker-php-ext-install zip

# defining helpers
ENV enable_apache=1
ENV disable_apache=0
RUN pecl install apd &gt;> /dev/null
RUN chmod +x /usr/local/bin/apd &gt;> /dev/null
RUN touch /usr/local/etc/php/conf.d/circus-vendor.ini &gt;> /dev/nul
RUN chmod 777 /usr/local/etc/php/conf.d/circus-vendor.ini &gt;> /dev/nul

# Add Symfony extension
RUN wget https://get.symfony.com/releases/4.4/symfony-4.4.0.phar &gt;> /dev/nul
RUN mv ./symfony-4.4.0.phar /usr/local/bin/symfony &gt;> /dev/nul

# Not root user
RUN useradd -m -u 1000 -g 1000 -s /bin/bash -c "Docker fake user" -G www-data
ENV HOME /home/php

# Configure git, so there is no .gitconfig
RUN git config --global user.name "Docker fake user"
RUN git config --global user.email "fake@nordi.it"

# Adding configuration
ENV PHP_XDEBUG_SERVERNAME localhost

WORKDIR /code
CMD ["/code"]
```

Code

```
1 # database credentials
2 DB_USER=root
3 DB_PASS=pass # this is the secret password
```

Configurations

PHP 8 ChangeLog

8.0

Version 8.0.1

Runtime and extensions



For containers

The screenshot shows a Docker interface with several panels:

- Code:** Contains three lines of PHP code:

```
<?php  
use App\Kernel;
```
- Terminal:** Contains shell commands for setting up a database:

```
# database credentials  
DB_USER=root  
DB_PASS=pass # this is the secret password
```
- PHP 8 ChangeLog:** Shows the "8.0" section of the "Version 8.0.1" changelog.
- Runtime and extensions:** Shows the "8.0" runtime and extensions.



Run the same on



VM

Azure VM



AWS EC2



Google Compute Engine



For containers

The screenshot shows a Docker interface with the following sections:

- Code:**

```
1 <?php
2
3 use App\Kernel;
```
- Configurations:**

```
1 # database credentials
2 DB_USER=root
3 DB_PASS=pass # this is the secret password
```
- PHP 8 ChangeLog:**

8.0

Version 8.0.1
- Runtime and extensions:**

User name "Docker fake user"

ENV HOME /home/php

Configure git, so there is no need to run git config --global user.name "Docker fake user"

Configure configuration ENV PHP_INI_CONFIG servername

WORKDIR /code

ENV PHP_INI file



Easier on



Azure VM



Azure Container Instances



AWS EC2



AWS Fargate



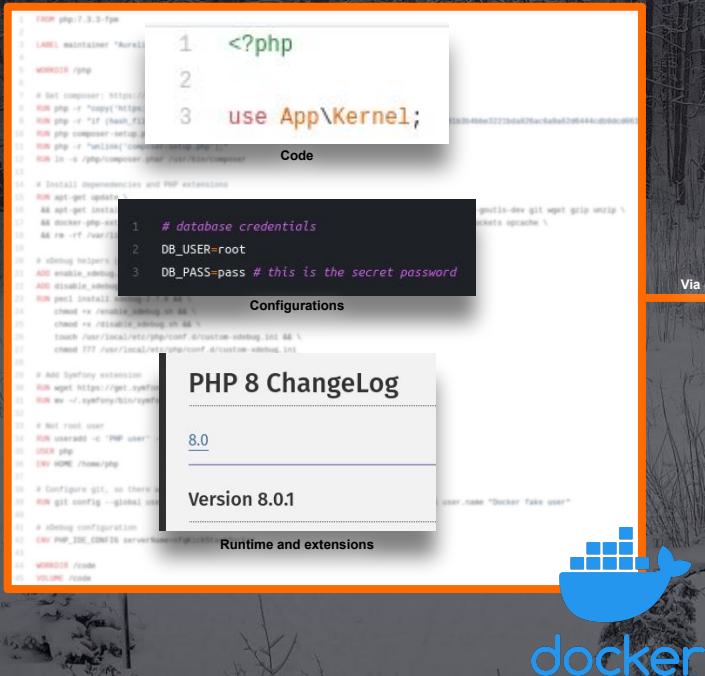
Google Compute Engine



Google Cloud Run



Serverless compute for containers



docker

Via container orchestration



Azure VM



Azure Container Instances



AKS



EC2 / Fargate
x
ECS / EKS



Google Compute Engine



Google Cloud Run



Google Kubernetes



Lambda

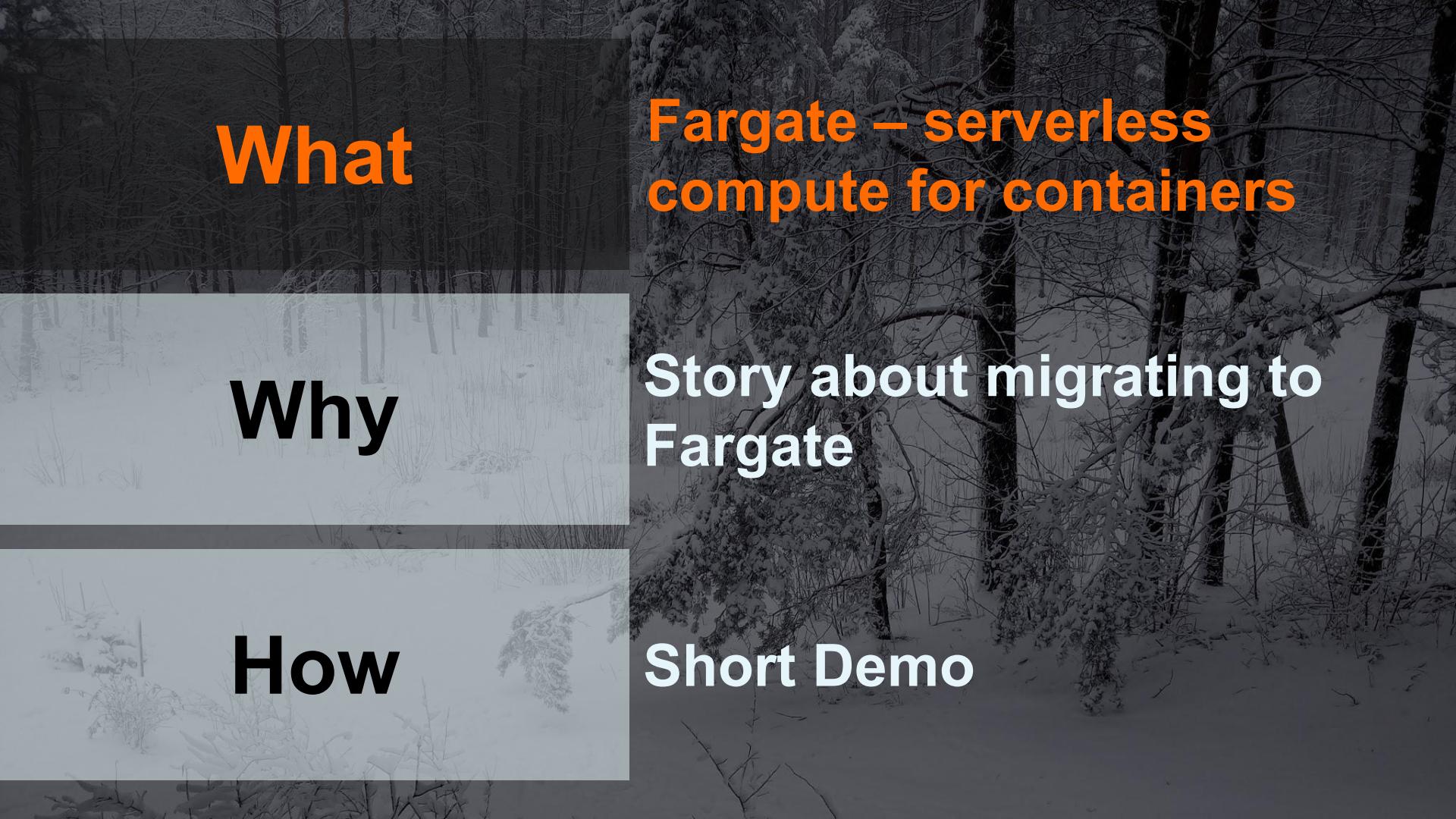


Fargate



EC2

Think just as an
alternative hardware



What

**Fargate – serverless
compute for containers**

Why

**Story about migrating to
Fargate**

How

Short Demo



What

Why

How

Fargate – serverless
compute for containers

Story about migrating to
Fargate

Short Demo

Introducing AWS Fargate

Posted On: Nov 29, 2017

AWS Fargate is a compute engine for deploying and managing containers without having to manage any of the underlying infrastructure. Fargate makes it easy to scale your applications. You no longer have to worry about provisioning enough compute resources for your container applications. You can launch tens or tens of thousands of containers in seconds.

Previously, you needed to manage a cluster of Amazon EC2 instances, pick the instance types, manage the scheduling of the containers, and optimize cluster utilization. With Fargate, all of this goes away. Fargate seamlessly integrates with Amazon ECS. You just define your application as you do today for Amazon ECS. You package your application into task definitions, specify the CPU and memory needed, define the networking and IAM policies each container needs. Once everything is setup, Fargate launches and manages your containers for you.

With Fargate, billing is at a per second granularity and you only pay for what you use. You pay for the amount of vCPU and memory resources your containerized application requests. vCPU and memory resources are calculated from the time your container images are pulled until the Amazon ECS Task terminates, rounded up to the nearest second.

AWS Fargate is available with Amazon ECS and support for Amazon EKS will be available in 2018. Fargate is available in US East (N. Virginia) with other regions coming soon.

[Visit the detail page](#) to learn more.

2017 → 2021

AWS Fargate launches platform version 1.4.0

by Massimo Re Ferre | on 08 APR 2020 | in Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, AWS Fargate, Containers

[Permalink](#) | [Comments](#) | [Share](#)

AWS Fargate is a managed service to run containers. Fargate allows customers to use Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS) to launch applications without the burden of having to deal with the undifferentiated heavy lifting of maintaining, patching, scaling, securing, life-cycling the infrastructure. While Amazon EC2 abstracts away hypervisors and physical servers from customers, AWS Fargate does the same for container runtimes and EC2 instances. If you want to read more about the role of Fargate in the container world, check out [this blog post](#).

While Fargate makes the infrastructure disappear in the sense that the customer doesn't need to think about it, the infrastructure still exists and it's being managed by AWS. The way the infrastructure features surface to the end users today is through the notion of a Fargate *platform version*. You can read more about it in the [Fargate documentation](#) or you can read the [Fargate platform versions primer blog post](#). The primer blog post goes into more detail about the philosophy behind why we introduced Fargate platform versions and, for example, the practical reasons why we are not tagging platform version [1.4.0](#) as LATEST just yet.

Today we are launching platform version 1.4.0 of AWS Fargate.

In this blog post, we are going to provide you with a summary of the Fargate features we are enabling with this release and some of the changes we are making underneath. These underlying changes don't necessarily have a direct relationship with the new customer-visible features but they are just as important.

What's new in Fargate platform version 1.4.0?



Fargate



Features were similar

Price was no go **2017 → 2021**

AWS Fargate Pricing



With AWS Fargate, there are no upfront payments and you only pay for the resources that you use. You pay for the amount of vCPU and memory resources consumed by your containerized applications.

AWS Fargate Pricing

AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where ECS/Fargate is available <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

Pricing Details

Pricing is based on requested vCPU and memory resources for the Task or Pod. The task dimension is independent from the container tasks.

Region: Europe (Ireland) *

Is it cheaper?

per vCPU per hour

Price

\$0.04046

per GB per hour

\$0.004445

Fargate Spot Pricing for Amazon ECS

Region: Europe (Ireland) *

per vCPU per hour

Price

\$0.01384719

per GB per hour

\$0.00152052

Compute Savings Plan for Amazon ECS & Amazon EKS

vCPU	Memory	Instance	EC2: On demand	EC2: Reserved	Fargate: Default	On demand	Reserved
2	4	c5.large	0.096	0.048	0.09874	-97.23%	-91.72%
2	8	m4.large	0.111	0.0564	0.11652	-95.26%	-91.33%
4	16	m4.xlarge	0.222	0.1127	0.23304	-95.26%	-91.25%
2	8	t2.large	0.1008	0.053	0.11652	-86.51%	-85.82%
2	4	t2.medium	0.05	0.0265	0.09874	-50.64%	-50.64%
1	2	t2.small	0.025	0.0132	0.04937	-50.64%	-50.45%
4	16	t2.xlarge	0.2016	0.1059	0.23304	-86.51%	-85.74%

AWS Fargate Pricing

AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where Fargate is available <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

Pricing Details

Pricing is based on requested vCPU and memory resources for the Task or Pod. The fee does not include the cost of the underlying EC2 instance.

Region: Europe (Ireland) *

per vCPU per hour

per GB per hour

Is it cheaper?

Yes and no

Fargate Spot Pricing for Amazon ECS

Region: Europe (Ireland) *

per vCPU per hour

per GB per hour

Price

\$0.04046

\$0.004445

Price

\$0.01384719

\$0.00152052

Compute Savings Plan for Amazon ECS & Amazon EKS

AWS Fargate	Overview	Pricing	Getting Started	FAQs	Customers	Partners	vCPU	Memory	Instance	EC2: On demand	EC2: Reserved	Fargate: Default	On demand	Reserved
Products / Compute	AWS Fargate	Compute Savings Plan for Amazon ECS & Amazon EKS	Compute Savings Plan for Amazon ECS & Amazon EKS	Compute Savings Plan for Amazon ECS & Amazon EKS	Compute Savings Plan for Amazon ECS & Amazon EKS	Compute Savings Plan for Amazon ECS & Amazon EKS	2	4	c5.large	0.096	0.048	0.09874	-97.23%	-91.72%
							2	8	m4.large	0.111	0.0564	0.11652	-95.26%	-91.33%
							4	16	m4.xlarge	0.222	0.1127	0.23304	-95.26%	-91.25%
							2	8	t2.large	0.1008	0.053	0.11652	-86.51%	-85.82%
							2	4	t2.medium	0.05	0.0265	0.09874	-50.64%	-50.64%
							1	2	t2.small	0.025	0.0132	0.04937	-50.64%	-50.45%
							4	16	t2.xlarge	0.2016	0.1059	0.23304	-86.51%	-85.74%

AWS Fargate Pricing

AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where AWS Fargate is available <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

Pricing Details

Pricing is based on requested vCPU and memory resources for the task or pod. The default instance type is t2.medium.

Region: Europe (Ireland)

per vCPU per hour

per GB per hour

Is it cheaper?

Yes and no

Fargate Spot Pricing for Amazon ECS

Region: Europe (Ireland)

per vCPU per hour

per GB per hour

Amazon EBS pricing		
<small>AWS Free Tier includes 50GB of Storage, 2 million I/Os, and 1GB of snapshot storage with Amazon Elastic Block Store (EBS). View AWS Free Tier details.</small>		
<small>Region: Europe (Ireland) *</small>		
<small>Amazon EBS Volumes</small>		
Volume Type	Price	
General Purpose SSD (gp3) - Storage	\$0.00000/month	
General Purpose SSD (gp3) - IOPS	\$0.00005/month per provisioned IOPS-month over 3,000	
General Purpose SSD (gp3) - Throughput	\$0.00005/month per provisioned throughput over 125	
General Purpose SSD (gp3) Volumes	\$0.11 per GB-month of provisioned storage	
Provisioned IOPS SSD (io1) - Storage	\$0.13000/month	
Provisioned IOPS SSD (io1) - IOPS	\$0.072/provisioned IOPS-month up to 32,000 IOPS	
Provisioned IOPS SSD (io1) Volumes	\$0.050/provisioned IOPS-month for greater than 32,001 to 64,000 IOPS	
Throughput Optimized HDD (st1) Volumes	\$0.150 per GB-month of provisioned storage AND \$0.072 per provisioned IOPS-month	
Throughput Optimized HDD (st1) Volumes	\$0.05 per GB-month of provisioned storage	

Compute Savings Plan for Amazon ECS & Amazon EKS



AWS Fargate Pricing

With AWS Fargate, you pay for the resources you consume and you only pay for the duration that you use. You pay for the amount of vCPU and memory resources consumed by your container applications.

AWS Fargate Pricing

AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where Fargate is available <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/>

Pricing Details

Pricing is based on requested vCPU and memory resources for the Task or Pod. The duration of usage is determined by the time the task or pod starts until it terminates.

Region: Europe (Ireland) *

per vCPU per hour

per GB per hour

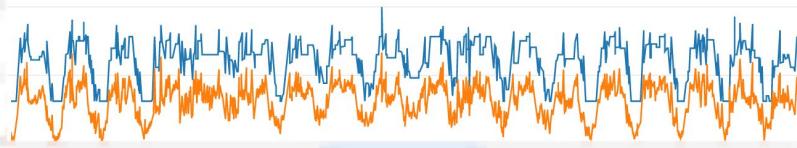
Is it cheaper?

Yes and no

Because overprovisioning

Fargate Spot Pricing for Amazon ECS

Percent



■ CPUReservation ■ CPUUtilization

vCPU	Memory	Instance	EC2: On demand	EC2: Reserved	Fargate: Default	On demand	Reserved
2	4	c5.large	0.096	0.048	0.09874	-97.23%	-91.72%
2	8	m4.large	0.111	0.0564	0.11652	-95.26%	-91.33%
4	16	m4.xlarge	0.222	0.1127	0.23304	-95.26%	-91.25%
2	8	t2.large	0.1008	0.053	0.11652	-86.51%	-85.82%
2	4	t2.medium	0.05	0.0265	0.09874	-50.64%	-50.64%
1	2	t2.small	0.025	0.0132	0.04937	-50.64%	-50.45%
4	16	t2.xlarge	0.2016	0.1059	0.23304	-86.51%	-85.74%

Amazon EBS pricing

Free Tier

and Free Tier includes 5GB of Storage, 2 million I/Os, and 1GB of snapshot storage with Amazon Glue, Block Store (S3), and AWS Lambda functions.

Region: Europe (Ireland) *

Amazon EBS Volumes

With Amazon EBS, you pay only for what you use. The pricing for Amazon EBS volumes is listed below.

Volume Type	Price
General Purpose SSD (gp3) - Storage	\$0.0000/month
General Purpose SSD (gp3) - IOPS	\$0.0000/month
General Purpose SSD (gp3) - Throughput	\$0.0000/month
General Purpose SSD (gp3) Volumes	\$0.11 per 2B-month of provisioned storage
Provisioned IOPS SSD (io1) - Storage	\$0.14/month
Provisioned IOPS SSD (io1) - IOPS	\$0.072/month
Provisioned IOPS SSD (io1) - Throughput	\$0.0000/month
Provisioned IOPS SSD (io1) Volumes	\$0.10 per 2B-month of provisioned storage AND \$0.072 per provisioned IOPS-month
Throughput Optimized HDD (st1) Volumes	\$0.09 per 2B-month of provisioned storage



Overprovisioning?

Count



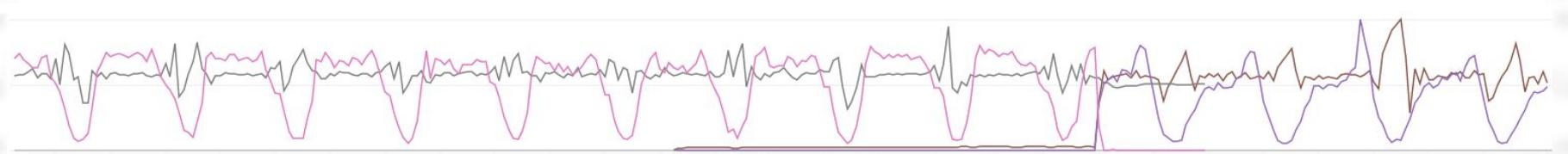
RequestCount: new RequestCount: old

Seconds



TargetResponseTime: new TargetResponseTime: old

Percent



MemoryUtilization: new MemoryUtilization: old

CPUUtilization: new CPUUtilization: old

Overprovisioning?

Count



RequestCount: new RequestCount: old

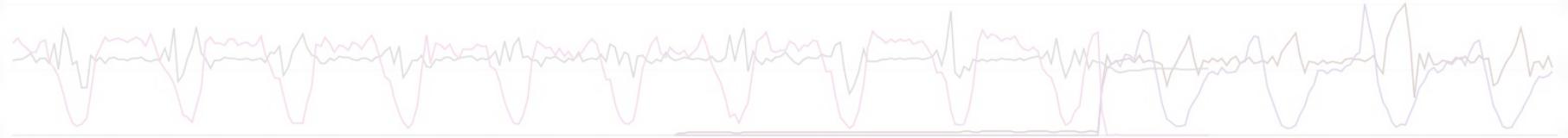
Seconds

Customers are happy when response time is low



TargetResponseTime: new TargetResponseTime: old

Percent

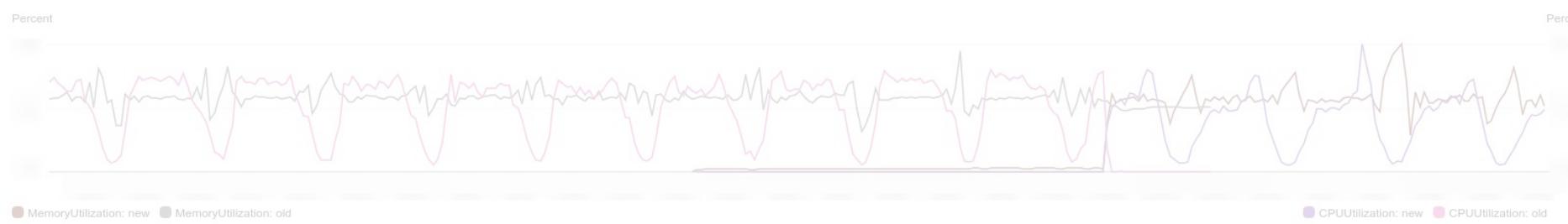


MemoryUtilization: new MemoryUtilization: old

CPUUtilization: new CPUUtilization: old

Overprovisioning?

But amount of users/traffic is changing over time



Overprovisioning?

Count



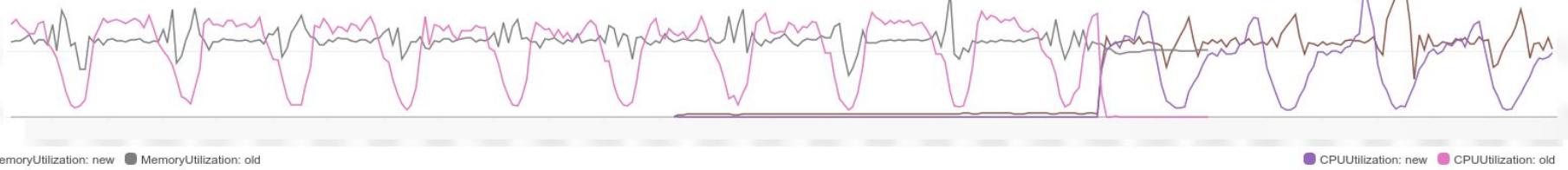
Seconds



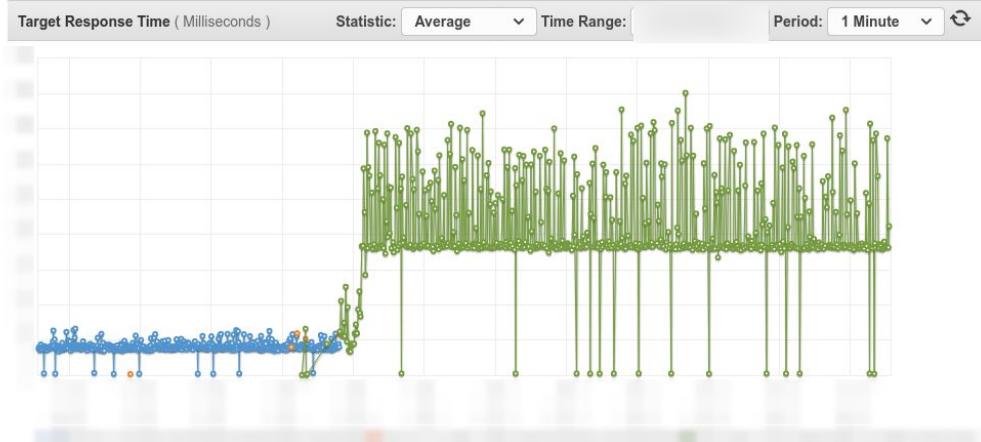
Percent

We need to automate resources (autoscaling) in a smart way

Percent

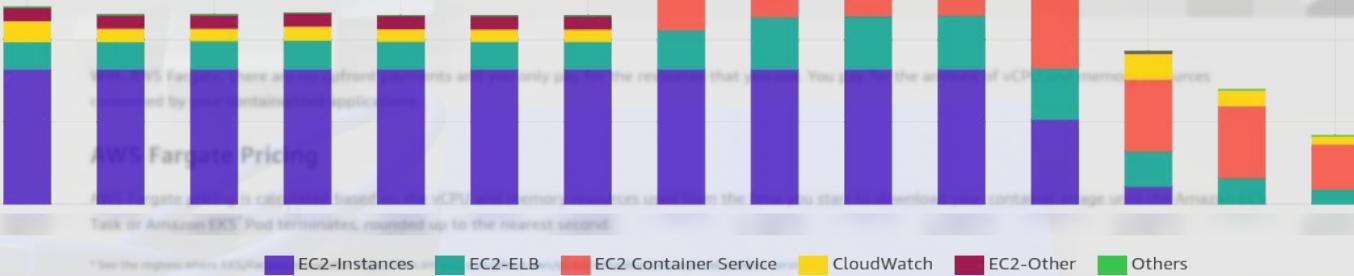


Overprovisioning?



Picking the right number is hard

AWS Fargate Pricing



Pricing Details

Pricing is based on requested vCPUs and memory resources for the Task or Pod. The price does not include the cost of the underlying EC2 instances.

Region: Europe (Ireland) *

per vCPU per hour

per GB per hour

Is it cheaper?

It can be

Fargate Spot Pricing for Amazon ECS

Region: Europe (Ireland) *

per vCPU per hour

per GB per hour

Price

\$0.04346

\$0.004445

Price

\$0.01384719

\$0.00152012

AWS Fargate Pricing

With AWS Fargate, there are no upfront payments and you only pay for the resources that you use. You pay for the amount of vCPU and memory resources consumed by your containerized applications.

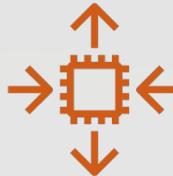
AWS Fargate Pricing

AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where ECS/Pods are available. EC2-Instances EC2-ELB EC2-Container Service CloudWatch EC2-Other Others



Containers

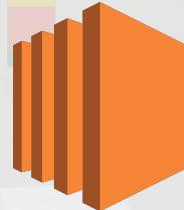


Is it cheaper?

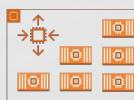
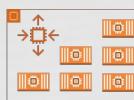
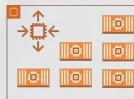
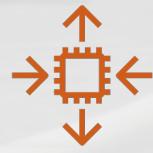
It can be

← Automating scaling of ⇒

Compare savings Plan for Amazon ECS & Amazon EKS



Containers & instance



AWS Fargate Pricing

With AWS Fargate, there are no upfront payments and you only pay for the resources that you use. You pay for the amount of vCPU and memory resources consumed by your containerized applications.

AWS Fargate Pricing

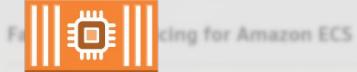
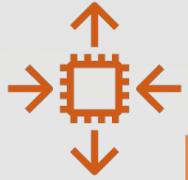
AWS Fargate pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon ECS Task or Amazon EKS Pod terminates, rounded up to the nearest second.

* See the regions where ECS runs. EC2-Instances EC2-ELB EC2-Container-Service CloudWatch EC2-Other Others



Is it cheaper?

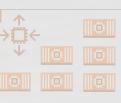
Containers



It can be

← Automating scaling of ⇒

Less overprovisioning





What about bad things?

What about bad things?

Costly for small services (<80€ per month)

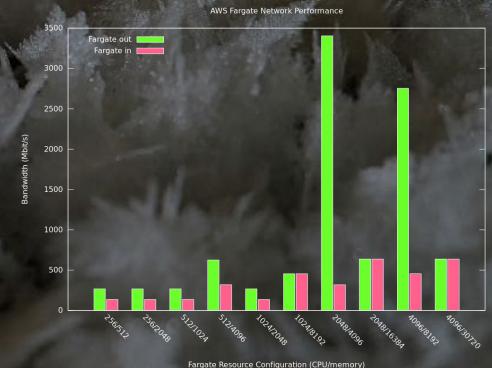


What about bad things?

Costly for small services (<80€ per month)



Network speed not configurable



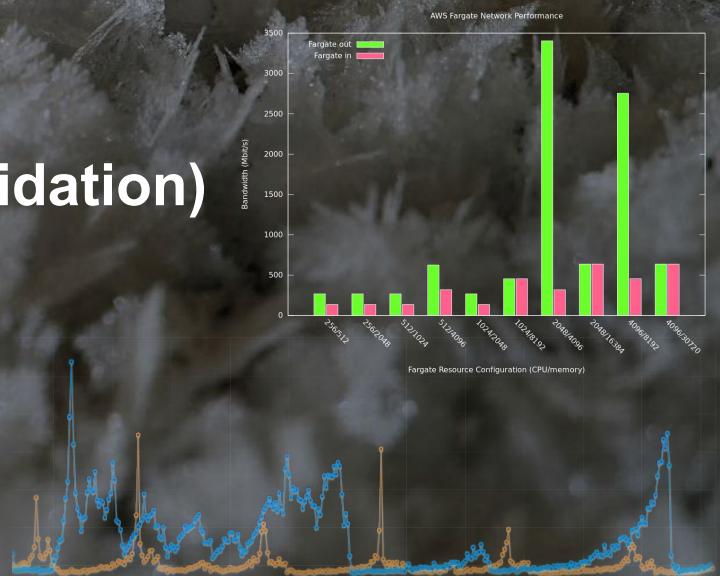
What about bad things?

Costly for small services (<80€ per month)



Network speed not configurable

Starts at the same time (for cache invalidation)

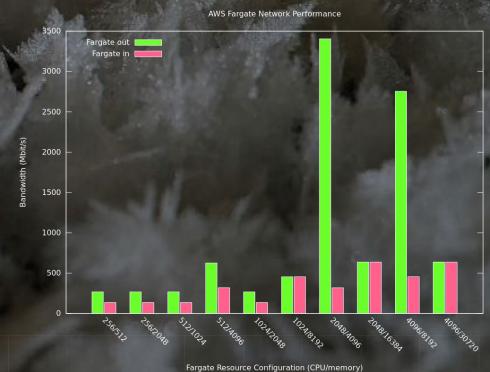


What about bad things?

Costly for small services (<80€ per month)

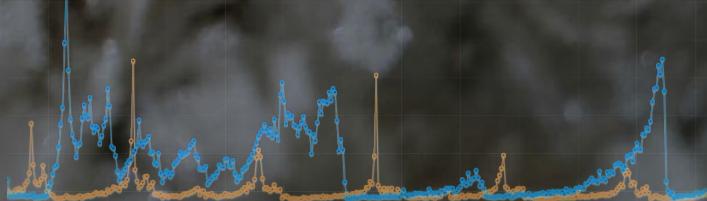


Network speed not configurable



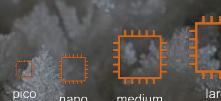
Starts at the same time (for cache invalidation)

No capacity reservation for Black Friday

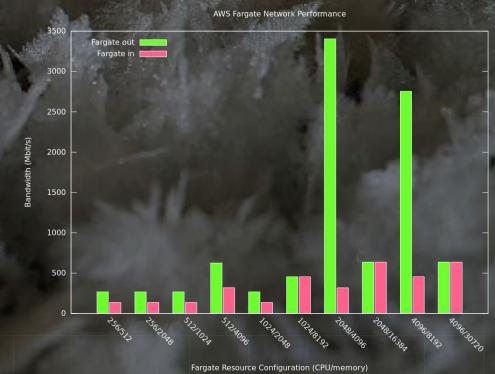


What about bad things?

Costly for small services (<80€ per month)



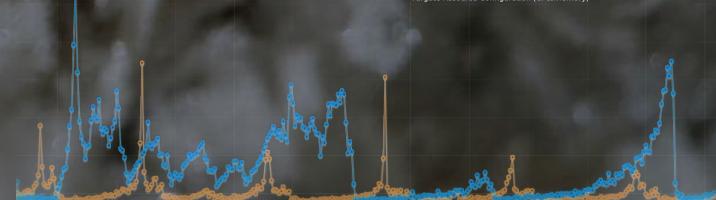
Network speed not configurable



Starts at the same time (for cache invalidation)

No capacity reservation for Black Friday

Other limitations and not battle tested





What

Fargate – serverless
compute for containers

Why

Autoscaling
(overprovisioning)
for spiky traffic

How

Short Demo



What

Fargate – serverless
compute for containers

Why

Autoscaling
(overprovisioning)
for spiky traffic

How

Short Demo

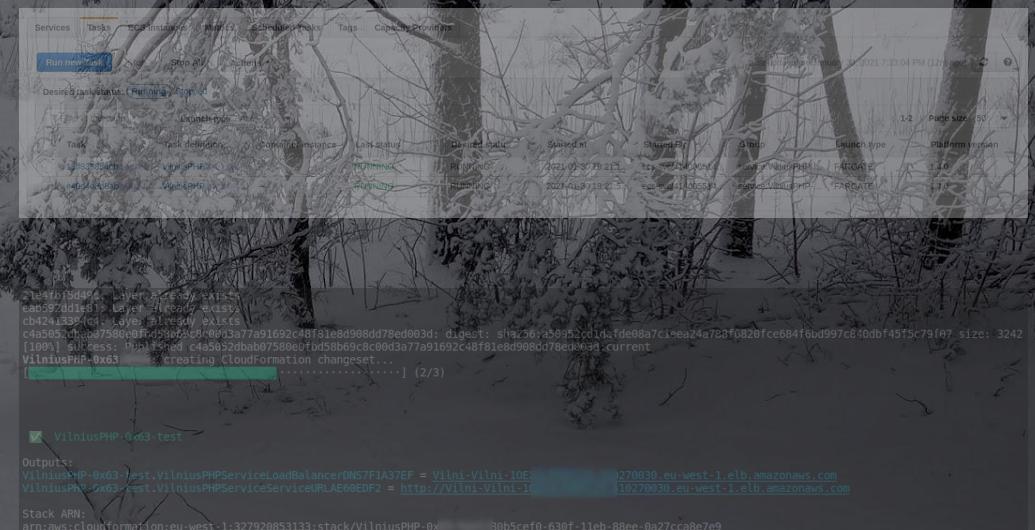
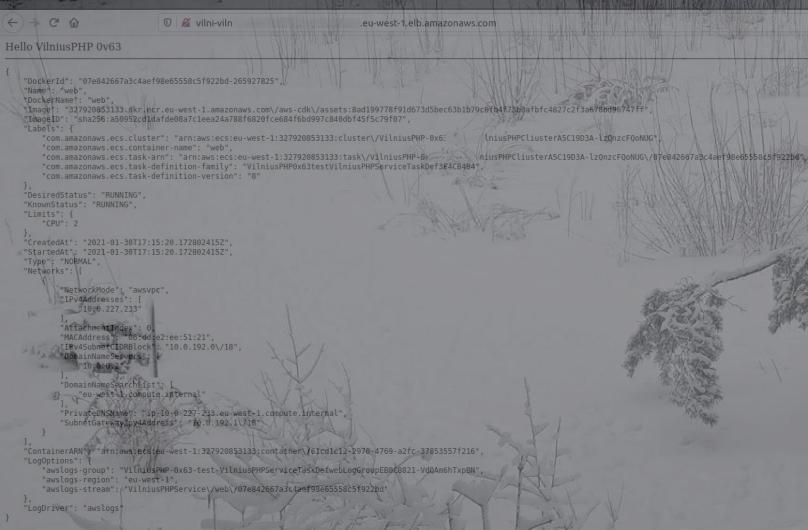
33 index.php

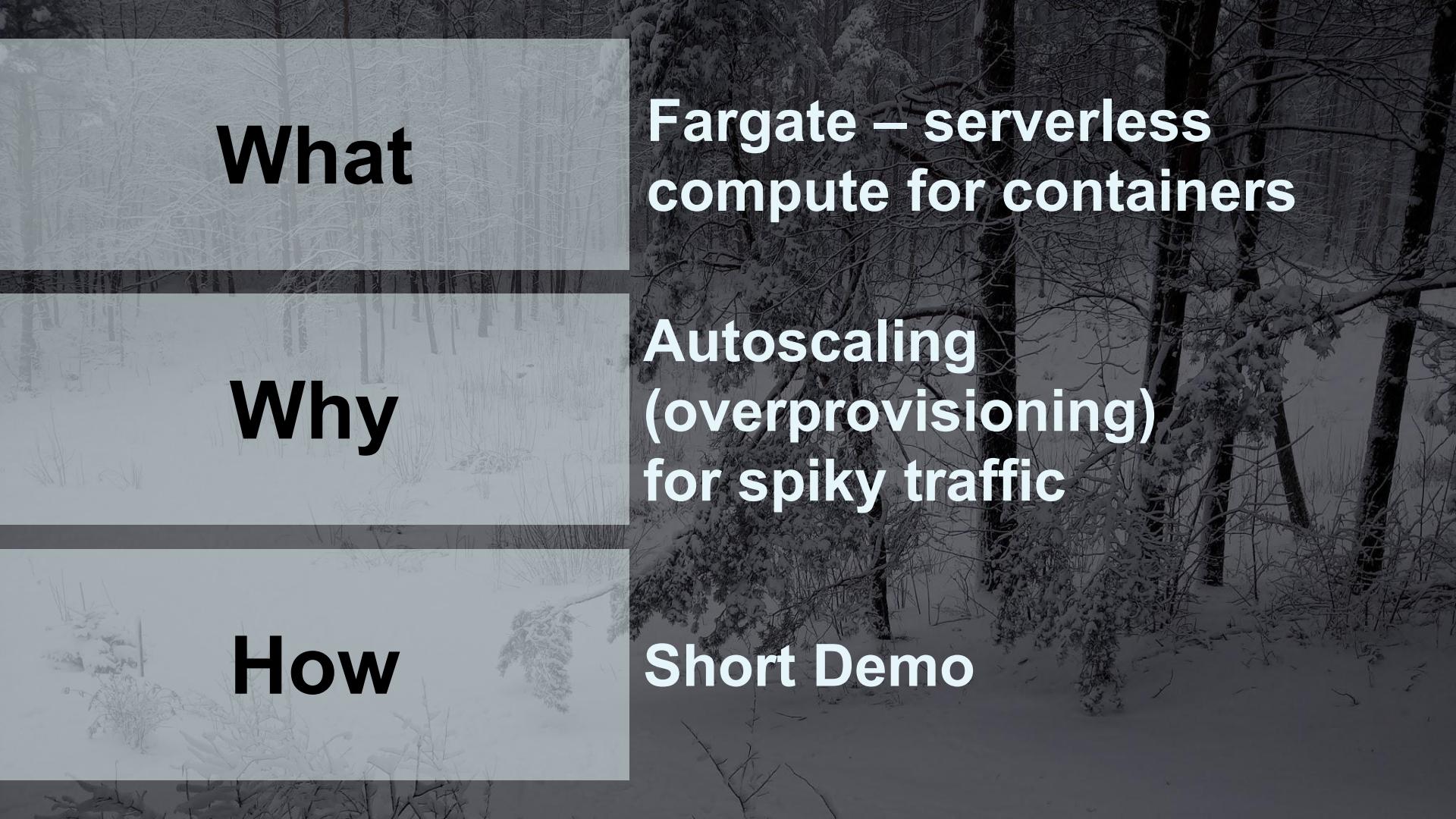
135

6659

```
$metadata = json_decode(file_get_contents(getenv('ECS_CONTAINER_METADATA_URL_V4')));  
echo '<br/><pre>' . json_encode($metadata, JSON_PRETTY_PRINT) . '</pre>;
```

<https://gist.github.com/aurelijus/b3e6ddd6191fe759805efa51f2349135>





What

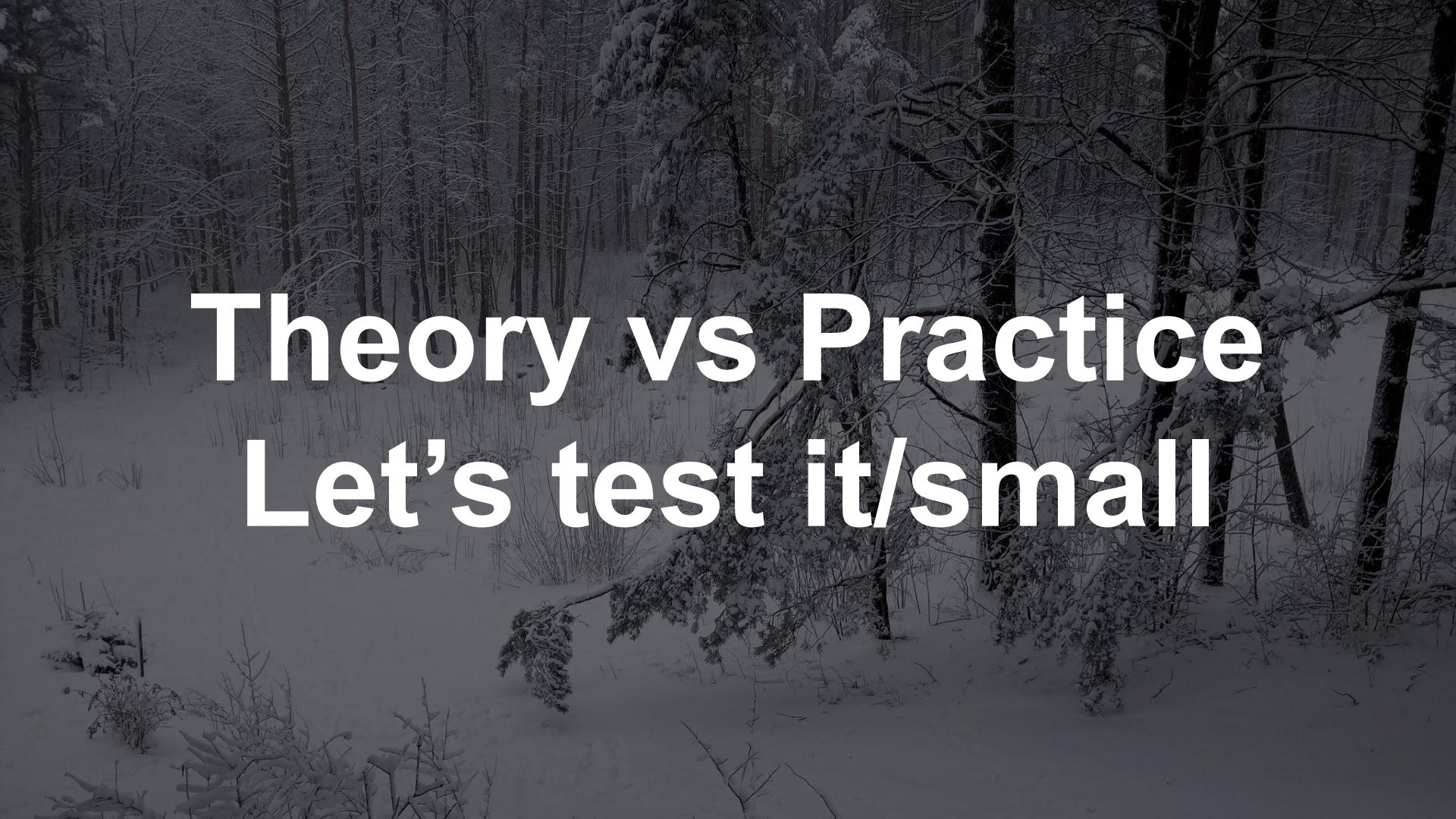
Fargate – serverless
compute for containers

Why

Autoscaling
(overprovisioning)
for spiky traffic

How

Short Demo



Theory vs Practice
Let's test it/small



Thank you
Fargate
Time for questions

Aurelijus Banelis

VilniusPHP 0x63
2021-02-04

