A 60-minute tour of AWS Compute

EC2 | Lambda | EC2 Container Service | Elastic Beanstalk

Julien Simon, Principal Technical Evangelist

@aws_actus @julsimon

Meetup AWS User Group Nantes #1

03/03/2016

AWS Compute technologies

EC2

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

Elastic Beanstalk

AWS Elastic Beanstalk is an application container for deploying and managing applications.

Lambda

AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources for you.

EC2 Container Service

Amazon ECS allows you to easily run and manage Docker containers across a cluster of Amazon EC2 instances.

Amazon EC2

- Infrastructure as a Service, launched in 2006
- Based on virtual machines ("EC2 instances") and images ("Amazon Machine Image", "AMI")
- Many instance types for different needs: general purpose, compute, memory, GPU, etc.
- Users can pick from Amazon-supported AMIs, vendor-supported AMIs ("EC2 Marketplace") or they can build their own

- All-inclusive: networking ("Virtual Private Cloud"), storage ("Elastic Block Storage"), firewalling ("Security Group"), load balancing ("Elastic Load Balancing"), high availability ("Availability Zones"), automatic scaling ("Auto-scaling groups"), monitoring ("Cloudwatch")
- Pay on an hourly basis

The best option if you need full control over your instances

Use Reserved Instances and Spot instances for massive savings

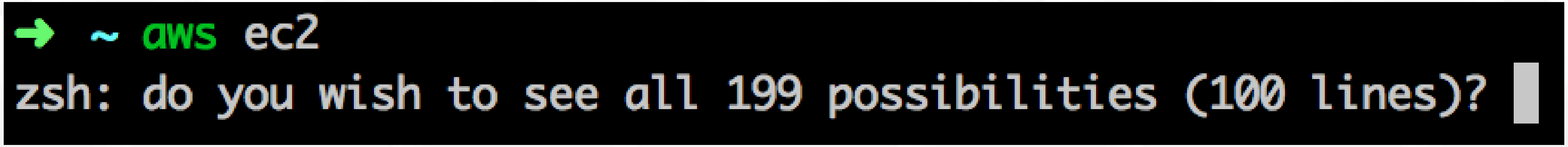
amazon web services

Amazon EC2 demo

Launch an Amazon Linux instance
in the default VPC with the default security group

```
$ aws ec2 run-instances --image-id ami-e1398992
--instance-type t2.micro --key-name mySshKey
--security-group-ids sg-09238e6d --region eu-west-1
```

This is the most important command ;)
Take some time to experiment with the '*aws ec2*' command line



```
➜  ~ aws ec2
zsh: do you wish to see all 199 possibilities (100 lines)?
```

Amazon Elastic Beanstalk

- Platform as a Service, launched in 2011
- Supports PHP, Java, .NET, Node.js, Python, Go, Ruby IIS, Tomcat and Docker containers

- Developer-friendly CLI : '*eb*'
- Uses AWS Cloudformation to build all required resources

- Built-in monitoring (Amazon Cloudwatch), networking (Amazon VPC), load balancing (Amazon ELB) and scaling (Auto-scaling)
- No charge for the service itself

- Relational data tier is available through Amazon Relational Data Service (RDS)

The simplest and most intuitive way to deploy your applications
This should really be your default option for deployment

Amazon Elastic Beanstalk demo

1. Create a new Rails application
2. Add a resource to the application
3. Declare a new Rails application in Amazon Elastic Beanstalk
4. Create an environment and launch the application

Create a new Rails application

```
$ rails new blog

$ cd blog

$ git add .

$ git commit -m "Initial version"
```

Create a new Rails application

```
$ git init

$ rails new blog

$ cd blog

$ git add .

$ git commit -m "Initial version"
```

Add a ‘post’ resource to the application

```
$ rails generate scaffold post title:string body:text

$ bundle exec rake db:migrate

$ git add .

$ git commit -m "Add post resource"

$ rails server

$ open http://localhost:3000/posts
```

Initialize a Ruby application

```
$ eb init blog -p Ruby -r eu-west-1

$ git add .gitignore

$ git commit -m "Ignore .elasticbeantalk directory"
```

Create a 'blog-dev' environment

Single instance (no auto-scaling, no load balancing),
t2.micro instance size (default value)

```
$ eb create blog-dev
--single
--keyname aws-eb
--envvars SECRET_KEY_BASE=`rake secret`

$ eb terminate blog-dev --force
```

Amazon EC2 Container Service

- Container as a Service, launched in 2015
- Built-in clustering, distributed state management, scheduling and high availability

- Developer-friendly CLI : ‘*ecs-cli*’
- Uses AWS Cloudformation to build all required resources
- Supports Docker 1.9.1, including Docker Compose files

- No charge for the service itself

A simple and scalable way to manage your Dockerized applications

Amazon ECS demo

```
$ ecs-cli configure --cluster myCluster --region eu-west-1
$ ecs-cli up --keypair lab2 --capability-iam --size 1
--instance-type t2.micro
$ ecs-cli compose service up

$ ecs-cli scale --size 3 --capability-iam
$ ecs-cli compose service scale 3

$ ecs-cli compose service delete
$ ecs-cli down myCluster --force
```

AWS Lambda

- Code as a Service, launched in 2014
- Supports Java, Python and Node.js

- Write and deploy pure functions to build event-driven, reactive applications
- Build APIs in conjunction with Amazon API Gateway
- Interact with other AWS services (S3, DynamoDB, etc)

- Pay as you go: number of requests + execution time (100ms slots)

The future: serverless applications and NoOps ☺

AWS Lambda demo

1. Write a simple Lambda function in Python
2. Create a REST API with API Gateway (resource + method)
3. Create a new stage
4. Deploy our API to the stage
5. Invoke the API with ‘*curl*’

A simple Lambda function in Python

```
def lambda_handler(event,context):
   result = event['value1'] + event['value2']
   return result
```

```
$ curl -H "Content-Type: application/json"
-X POST -d "{\"value1\":5, \"value2\":7}" https://API_ENDPOINT/STAGE/RESOURCE
12%
```

And now the trip begins. Time to explore!

Thank you. Let's keep in touch!

@aws_actus @julsimon
facebook.com/groups/AWSFrance/

AWS User Groups in Paris,
Lyon, Nantes, Lille & Rennes
(meetup.com)

March 7-8

AWSome Day

March 16

ioT world

March 23-24

sido
The Connected Business

April 6-7 (Lyon)

DEVOXX France

April 20-22

dot Scale

April 25

amazon web services

AWS Summit
May 31st

amazon web services