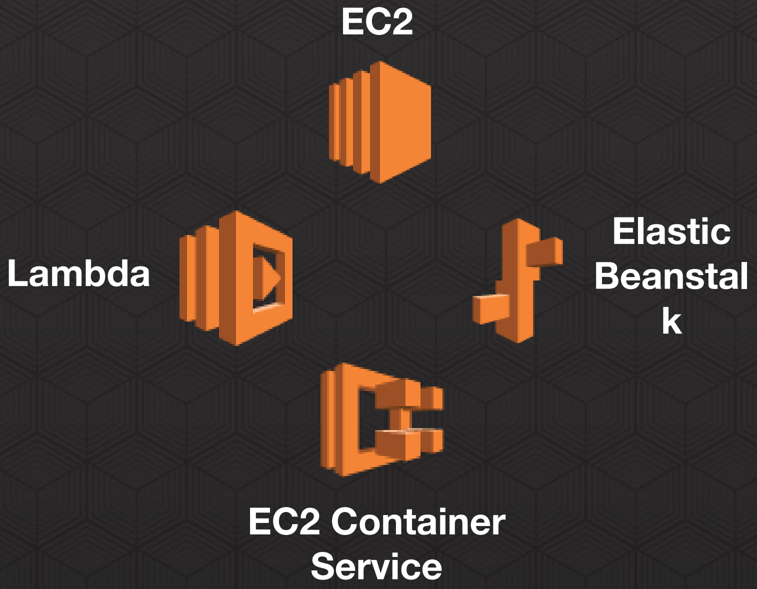
A 60-mn tour of AWS Compute

Julien Simon
Principal Technical Evangelist
Amazon Web Services

julsimon@amazon.com
@julsimon

08/11/2016



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.

https://aws.amazon.com/fr/blogs/compute/

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

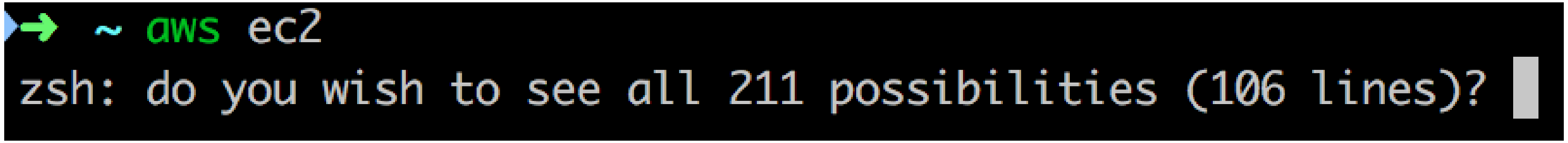
Use Reserved Instances and Spot Instances for massive savings (up to 90%)

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 aws-eb
--security-group-ids sg-d9906fbe --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 211 possibilities (106 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)
- Relational data tier is available through Amazon Relational Data Service (RDS)
- No charge for the service itself

The simplest and most intuitive way to deploy your applications

This should really be your default option for deployment

Supported platforms

http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts.platforms.html

docker-1.11.2
docker-1.6.2
docker-1.7.1
docker-1.9.1
multi-container-docker-1.11.2-(generic)
multi-container-docker-1.6.2-(generic)

glassfish-4.0-java-7-(preconfigured-docker)
glassfish-4.1-java-8-(preconfigured-docker)

java-7
java-8

tomcat-6
tomcat-7
tomcat-7-java-6
tomcat-7-java-7
tomcat-8-java-8

go-1.3-(preconfigured-docker)
go-1.4
go-1.4-(preconfigured-docker)
go-1.5

iis-7.5
iis-8
iis-8.5

node.js

php-5.3
php-5.4
php-5.5
php-5.6
php-7.0

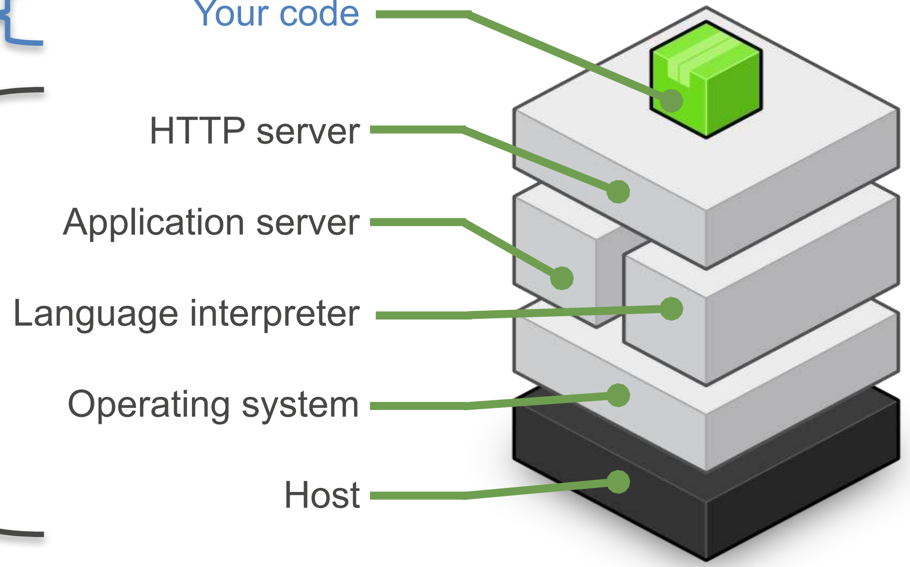
python-2.7
python-3.4
python-3.4-(preconfigured-docker)

ruby-1.9.3
ruby-2.0-(passenger-standalone)
ruby-2.0-(puma)
ruby-2.1-(passenger-standalone)
ruby-2.1-(puma)
ruby-2.2-(passenger-standalone)
ruby-2.2-(puma)
ruby-2.3-(passenger-standalone)
ruby-2.3-(puma)

ElasticBeanstalk vs. DIY

Focus on building your application

Elastic Beanstalk configures each EC2 instance in your environment with the components necessary to run applications for the selected platform. No more worrying about logging into instances to install and configure your application stack.



Amazon Elastic Beanstalk demo

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

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 deploy

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

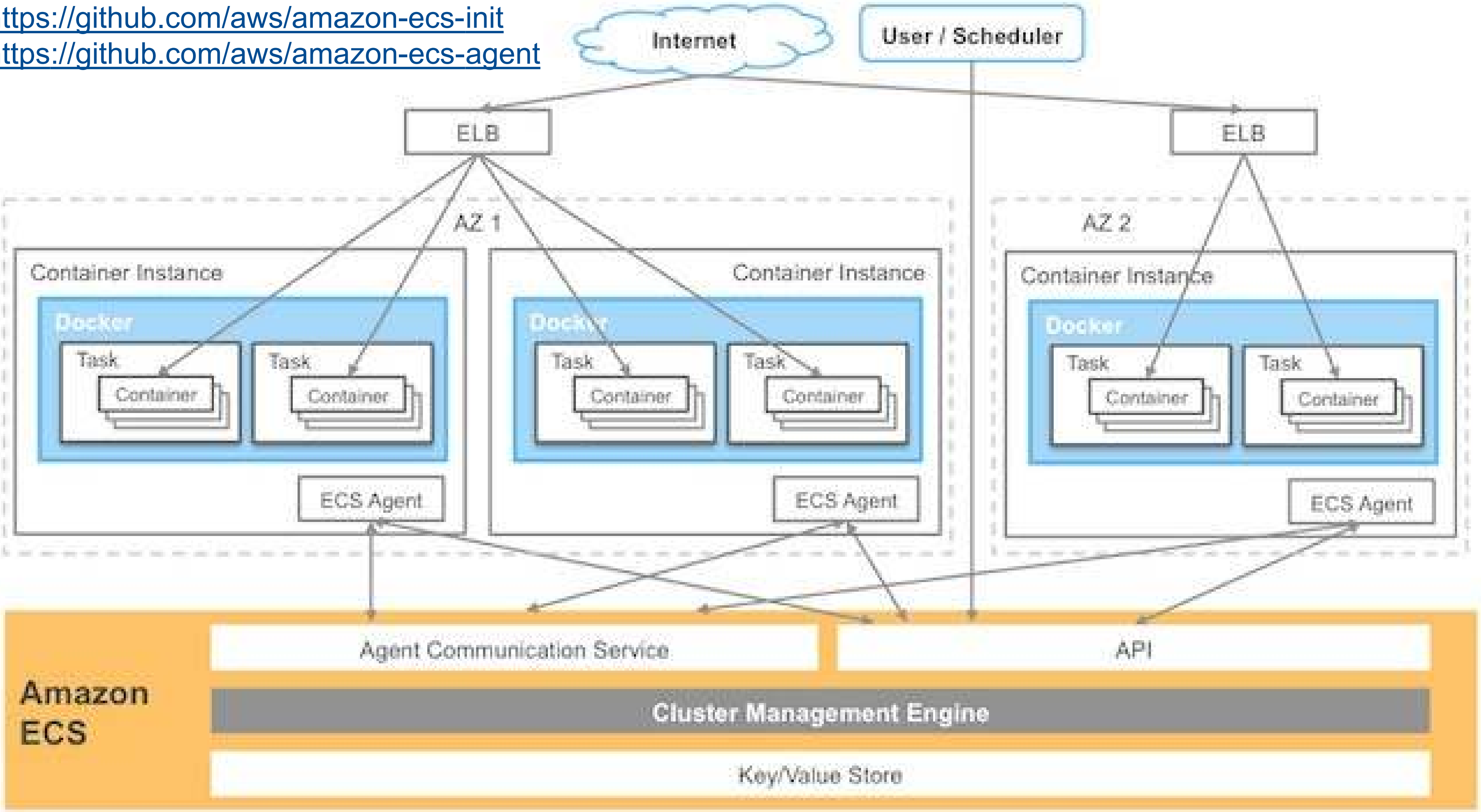
Amazon EC2 Container Service

- Container as a Service, launched in 2015
- Built-in clustering, distributed state management, scheduling and high availability
- Amazon EC2 Container Registry (ECR) for image storage

- Developer-friendly CLI : '*ecs-cli*'
- Uses AWS Cloudformation to build all required resources
- No charge for the service itself

A simple and scalable way to manage your Dockerized applications

https://github.com/aws/amazon-ecs-init
https://github.com/aws/amazon-ecs-agent



http://www.allthingsdistributed.com/2015/07/under-the-hood-of-the-amazon-ec2-container-service.html

Amazon ECS demo

Simple PHP application hosted in Amazon ECR + Docker Compose file

```
$ ecs-cli configure --cluster myCluster --region eu-west-1
$ ecs-cli up --keypair aws-eb --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
```

Homemade tool: ‘ecs-find’
https://github.com/juliensimon/aws/blob/master/ecs/ecs-find

AWS Lambda

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

- Write and deploy pure functions to build event-driven 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

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
```

```
aws lambda create-function --function-name myFunc \
--handler myFunc.lambda_handler --runtime python2.7 \
--zip-file fileb://myFunc.zip --memory-size 128 \
--role arn:aws:iam::ACCOUNT_NUMBER:role/lambda_basic_execution

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

AWS Lambda with the Serverless framework

http://github.com/serverless/serverless

- Run/test AWS Lambda functions locally, or remotely
- Auto-deploys & versions your Lambda functions
- Auto-deploys your REST API to AWS API Gateway
- Auto-deploys your Lambda events
- Support for multiple stages
- Support for multiple regions within stages
- Manage & deploy AWS CloudFormation resources

Other notable frameworks: Chalice (Python), Zappa (Python), Apex (golang)

And now the trip begins. Time to explore!

Going further with Amazon ECS

Tech articles by Werner Vogels, CTO, Amazon.com

http://www.allthingsdistributed.com/2014/11/amazon-ec2-container-service.html

http://www.allthingsdistributed.com/2015/04/state-management-and-scheduling-with-ecs.html

http://www.allthingsdistributed.com/2015/07/under-the-hood-of-the-amazon-ec2-container-service.html

Amazon ECS videos @ AWS re:Invent 2015

Amazon ECS: Distributed Applications at Scale https://www.youtube.com/watch?v=eun8CqGqdk8

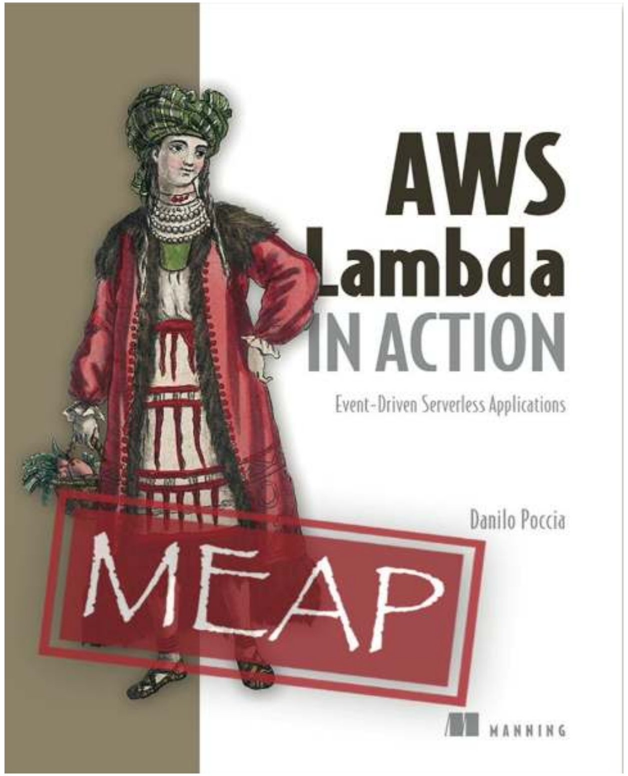
Turbocharge Your Deployment Pipeline with Containers https://www.youtube.com/watch?v=o4w8opVCI-Q

From Local Docker Development to Production https://www.youtube.com/watch?v=7CZFpHUPqXw

Going further with AWS Lambda

AWS re:Invent 2014 | (MBL202) NEW LAUNCH: Getting Started with AWS Lambda
https://www.youtube.com/watch?v=UFj27laTWQA
AWS re:Invent 2015 | (DEV203) Amazon API Gateway & AWS Lambda to Build Secure and Scalable APIs
https://www.youtube.com/watch?v=ZBxWZ9bgd44
AWS re:Invent 2015 | (DVO209) JAWS: The Monstrously Scalable Serverless Framework
https://www.youtube.com/watch?v=D_U6luQ6I90
https://github.com/serverless/serverless
AWS re:Invent 2015 | (ARC308) The Serverless Company Using AWS Lambda
https://www.youtube.com/watch?v=U8ODkSCJpJU
AWS re:Invent 2015 | (CMP407) Lambda as Cron: Scheduling Invocations in AWS Lambda
https://www.youtube.com/watch?v=FhJxTIq81AU
Reference architectures
http://www.allthingsdistributed.com/2016/06/aws-lambda-serverless-reference-architectures.html

Upcoming book on AWS Lambda



Written by AWS Technical Evangelist Danilo Poccia

Early release available at:

https://www.manning.com/books/aws-lambda-in-action

More sessions

- ~~7/11, 15:00 Hands-on with AWS IoT~~
- ~~8/11, 10:00 A 60-minute tour of AWS Compute~~
- 9/11, 10:00 Deep Dive: DevOps on AWS
- 9/11, 11:00 Running Docker clusters on AWS

- 21/11, 11:00 Move fast, build things with AWS
- 22/11, 11:00 Deep Dive: Amazon RDS

Danke sehr!

Julien Simon
Principal Technical Evangelist
Amazon Web Services

julsimon@amazon.com
@julsimon