



Amazon Web Services & Docker

Primo Docker Meetup - Milano 11 Dicembre 2014



Paolo Latella

XPeppers
Cloud Solutions Architect

AWS User Group Italia
Co-Founder

Paolo.latella@xpeppers.com

@latellapaolo

Topics

- ▣ Amazon Web Services
 - ▣ Deployment and management
- ▣ Elastic Beanstalk
 - ▣ Elastic Beanstalk & Docker
 - ▣ Demo
- ▣ Elastic Container Services (ECS)

Amazon Web Services



*Delivery of IT resources and applications by
API with pay-as-you-go pricing.*

AWS Deployment & Management

All AWS Services

Compute

Storage & Content Delivery

Database

Networking

Administration & Security

Analytics

Application Services

Deployment & Management >

Mobile Services

Enterprise Applications

Elastic Beanstalk

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

OpsWorks

AWS OpsWorks is a DevOps platform for managing applications of any scale or complexity on the AWS cloud.

CloudFormation

AWS CloudFormation lets you create and update a collection of related AWS resources in a predictable fashion.

CodeDeploy

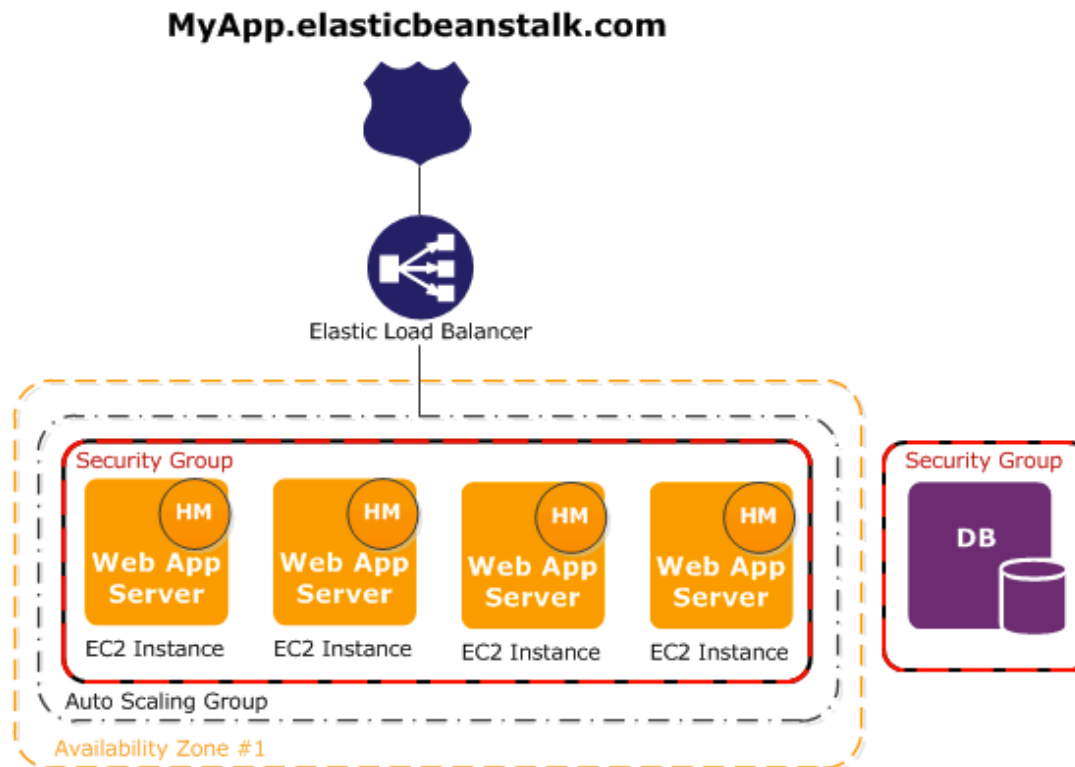
AWS CodeDeploy lets you fully automate code deployments.



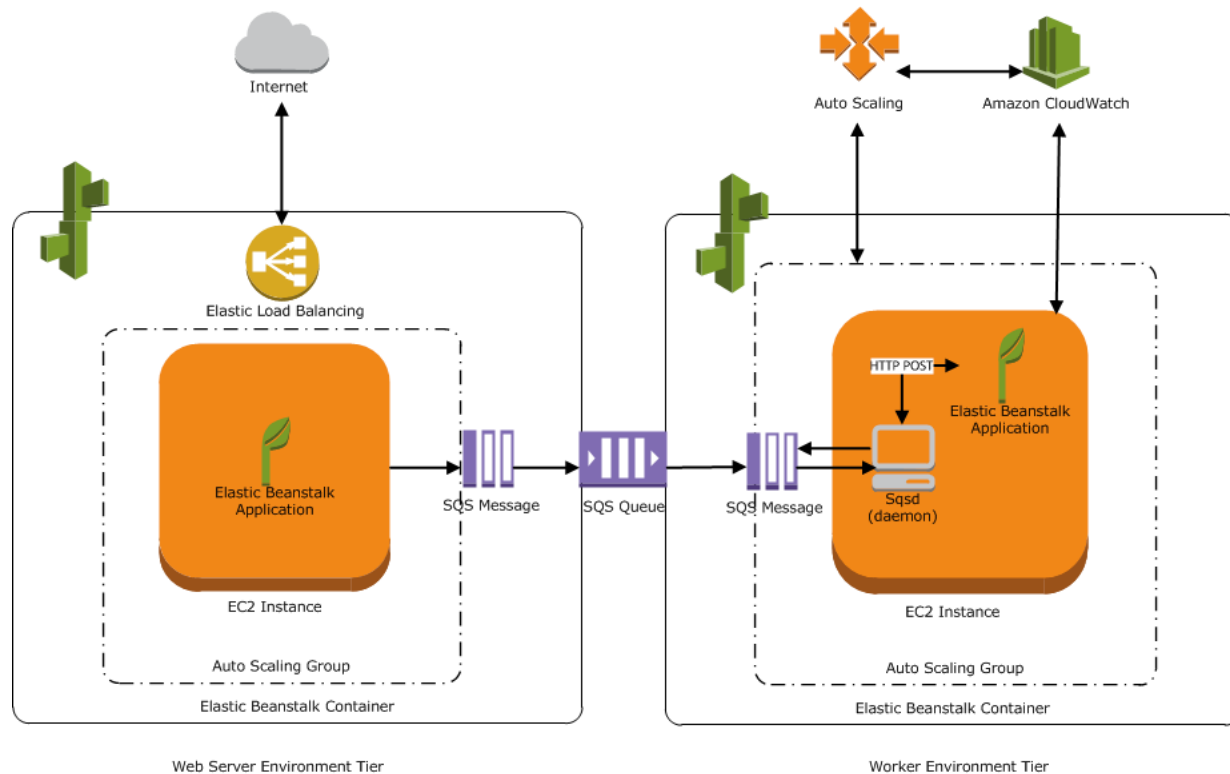
Elastic Beanstalk

- ❑ Possiamo fare il deploy delle applicazioni in maniera rapida e semplice senza preoccuparci dell'infrastruttura.
 - ❑ Riduce le complessità lasciando comunque libertà di azione (posso sempre accedere alle risorse dalla console)
- ❑ Environment
 - ❑ Default Tier (Web Server e Workers) or custom
- ❑ Platform
 - ❑ Java, .Net, PHP, etc.
- ❑ Provisioning, load balancing, scaling and monitoring fully managed by AWS Elastic Beanstalk

Environments: Web Server Tiers



Environments: Workers Tiers



Platforms

- ❑ The software stack on EC2 instances depends on selected platform
 - ❑ **Docker container**
 - ❑ Java (Java + Tomcat + Apache)
 - ❑ .Net (IIS)
 - ❑ Node.JS (Node.JS + Nginx or Apache)
 - ❑ PHP (PHP + Apache)
 - ❑ Python (Python + Apache + mod_wsgi)
 - ❑ Ruby (Ruby + Nginx + Passenger/Puma)

Elastic Beanstalk & Docker: demo



Elastic Beanstalk & Docker: dockerfile

```
FROM ubuntu:14.04
# Ubuntu and nodeJS for ElasticBeanstalk
#
# VERSION          0.0.1

FROM ubuntu:14.04
MAINTAINER Paolo Latella <paolo.latella@xpeppers.com>

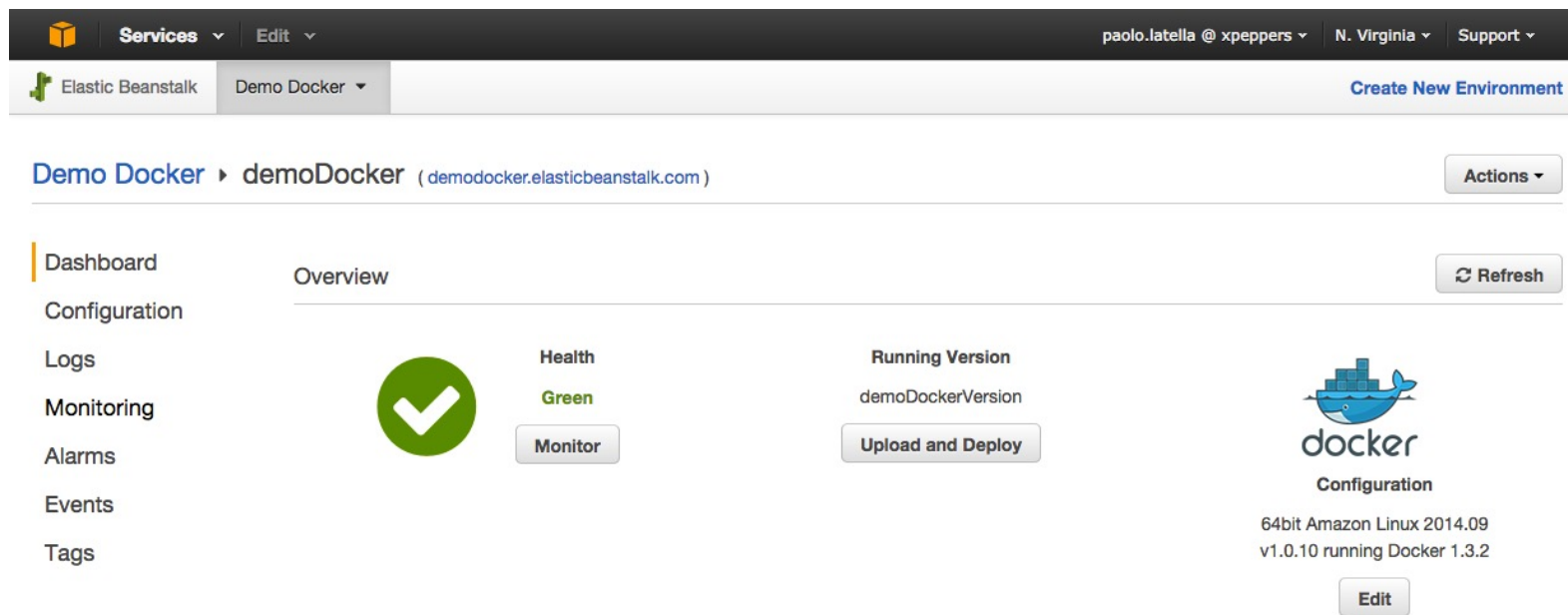
#Port mapping
EXPOSE 8080

#Update and install nodejs
RUN apt-get update && apt-get install -y nodejs

#Copy files for nodejs application
RUN mkdir /var/www/
ADD myws.js /var/www/

#Start application
CMD /usr/bin/nodejs /var/www/myws.js
```

Elastic Beanstalk & Docker: console



The screenshot displays the AWS Elastic Beanstalk console interface. At the top, the navigation bar shows 'Services' and 'Edit' dropdowns, along with the user 'paolo.latella @ xpeppers' and the region 'N. Virginia'. Below this, the 'Elastic Beanstalk' service is selected, and the 'Demo Docker' environment is chosen from a dropdown. A 'Create New Environment' button is visible on the right.

The main content area shows the 'demoDocker' environment (demodocker.elasticbeanstalk.com) with an 'Actions' dropdown. On the left, a sidebar lists navigation options: Dashboard, Configuration, Logs, Monitoring, Alarms, Events, and Tags.

The 'Overview' tab is active, displaying a large green checkmark icon. To the right of the icon, the 'Health' status is 'Green', with a 'Monitor' button. Further right, the 'Running Version' is 'demoDockerVersion', with an 'Upload and Deploy' button. On the far right, the 'docker' logo is shown above the 'Configuration' section, which lists '64bit Amazon Linux 2014.09 v1.0.10 running Docker 1.3.2' and includes an 'Edit' button. A 'Refresh' button is located at the top right of the overview section.

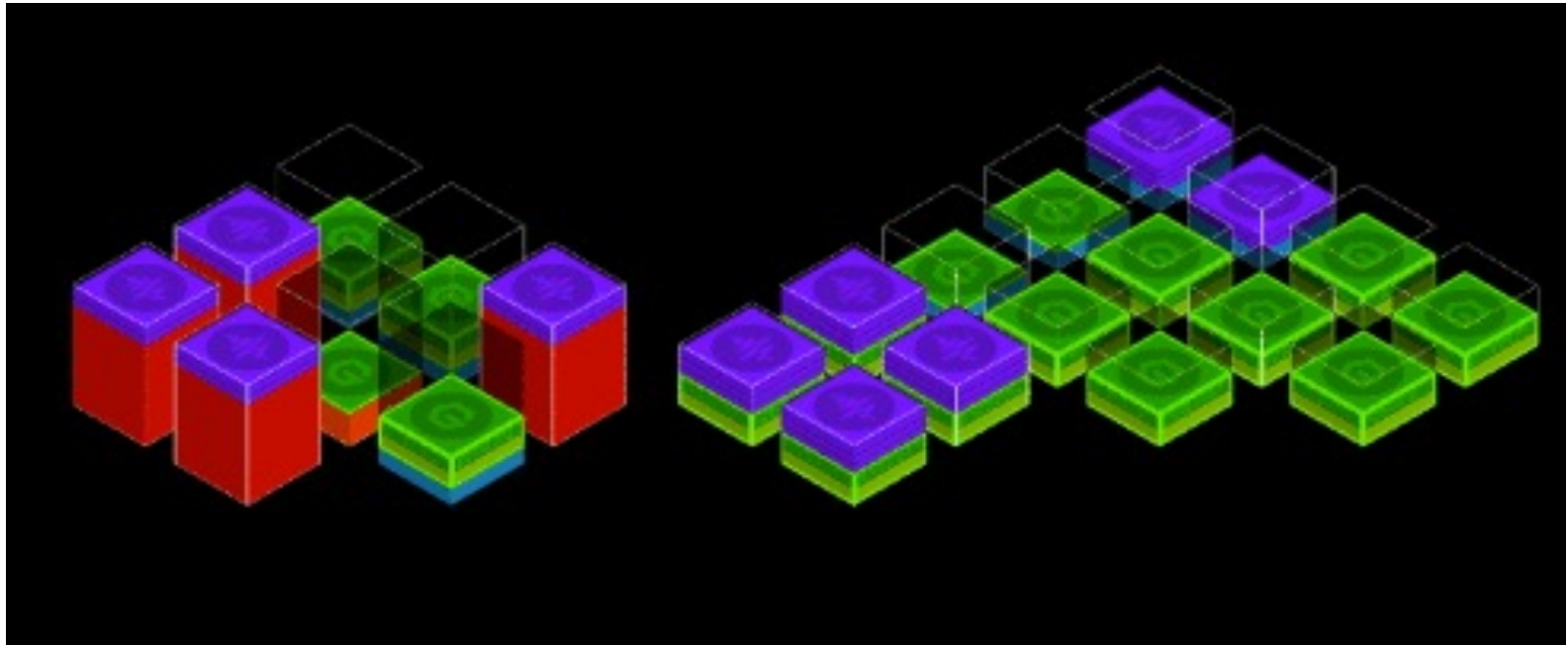
Elastic Container Services

- ❑ Cluster of container on EC2 instances
 - ❑ Configuration management and Monitoring fully managed by ECS
- ❑ Docker compliant
 - ❑ EC2 instance has a Docker agent and run one or more containers
- ❑ Task e Scheduler
 - ❑ Ttask (Json file) define how the container work on task, the computing resources (CPU and RAM) and the dependency
 - ❑ The scheduler launch the on Cluster with goal to optimize the resources of instances.

Elastic Container Services: task

```
task.json
1  {
2    "family": "DemoDockerMeetup-Milano",
3    "version": "1.0",
4    "containers": [
5      {
6        "name": "webserver",
7        "image": "nginx: latest",
8        "cpu": 512,
9        "memory": 128,
10       "portmapping": [
11         {
12           "containerport": 9443,
13           "hostport": 443
14         }
15       ],
16       "links": [
17         "rails",
18         "dbms"
19       ]
20     },
21     {
22       "name": "rails"
23       ...
24     },
25     {
26       "name": "dbms",
27       "image": "mysql: latest",
28       "cpu": 512,
29       "memory": 128,
30       "portmapping": [
31         {
32           "containerport": 3306,
33           "hostport": 3306
34         }
35       ]
36     }
37   ]
38 }
39 }
```

Elastic Container Services



Riferimenti

Application Management

<http://aws.amazon.com/application-management/>

Elastic Beanstalk

<http://aws.amazon.com/documentation/elastic-beanstalk/>

Amazon ECS

<http://aws.amazon.com/ecs/>

Deploy docker on AWS

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

Grazie!

