AWS

SUMMIT

Deploy a Deep Learning Framework on Amazon ECS and EC2 Spot Instances

Chad Schmutzer, AWS Solutions Architect
Hong Pham, AWS Solutions Architect
Patrick Shumate, AWS Solutions Architect
Tipu Qureshi, AWS Cloud Support Engineer

July 26, 2017

amazon web services

Before we get started

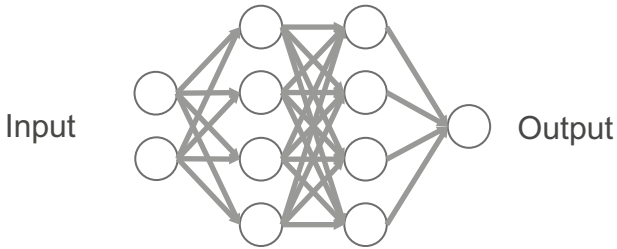
- Is everyone connected to WiFi? Power?
- You will be using your own AWS Account in this workshop
- Does everyone have the credits?
- If your AWS account is < 24 hours old, or if you have never launched an EC2 instance in your account, **please raise your hand and provide your AWS Account # to one of us**
- Please don’t forget to complete the evaluation in the app

What to expect from this workshop

- Hands-on, self-paced workshop
- Introduce MXNet
- Containers
- Overview of Amazon ECS & Amazon ECR
- Overview of AWS CloudFormation
- Overview of EC2 Spot instances
- Why use ECS and EC2 Spot Instances together
- Most importantly- work together and have some fun!

What is MXNet?

- Open-source deep learning framework - https://github.com/dmlc/mxnet
- Define, train, and deploy deep neural networks
- Highly scalable – single/multiple hosts, CPU/GPU support
- Support for multiple languages



dmlc mxnet

Why containers?

- Increase infrastructure utilization
- Environment isolation and fidelity
- Run diverse applications on shared hardware
- Changes are tracked
- Easy to deploy
- Buzzword…Microservices

Containers increase agility

Portability

Same ommutable images. Run anywhere.

Flexibility

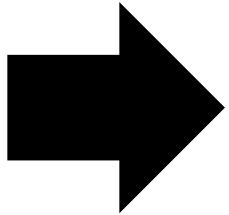
Create modular environments. Decompose apps.

Speed

Speeds up build and release cycle.

Efficiency

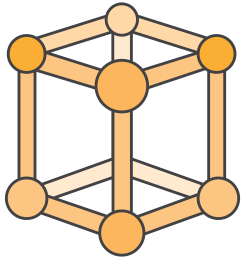
Optimize resource utilization.



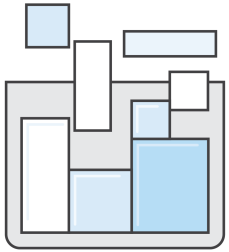
Agility

Amazon EC2 Container Service (ECS) & Amazon EC2 Container Registry (ECR)

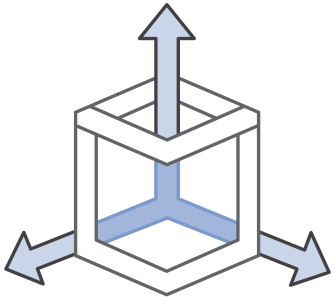
ECS benefits



Cluster management made easy

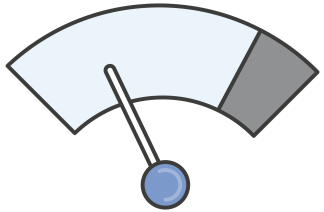


Flexible scheduling



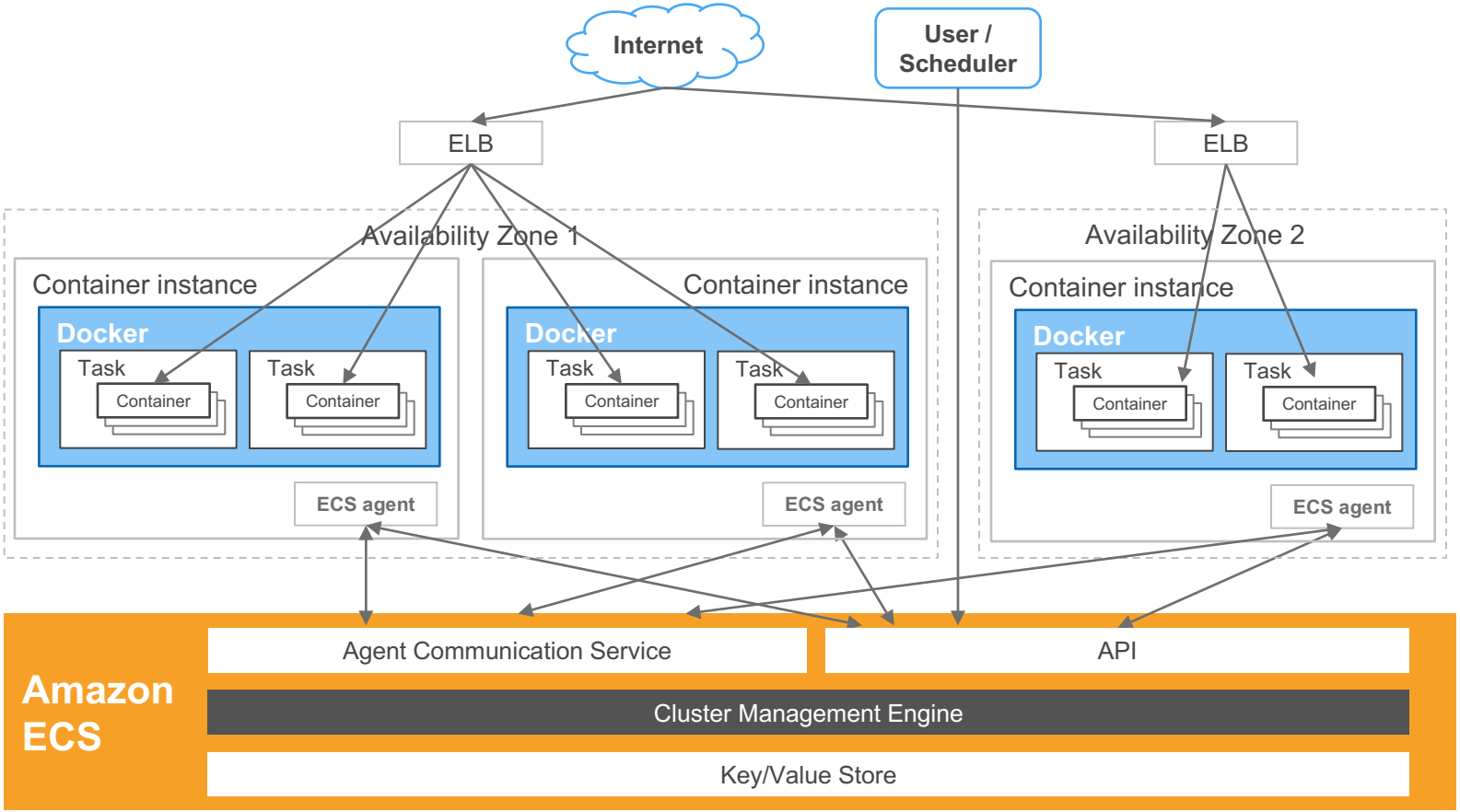
Integrated and extensible

Security



Performance at scale

ECS architecture



What Is ECR?

- Amazon EC2 Container Registry (ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon EC2 Container Service (ECS), simplifying your development to production workflow.
- Learn More: https://aws.amazon.com/ecr/

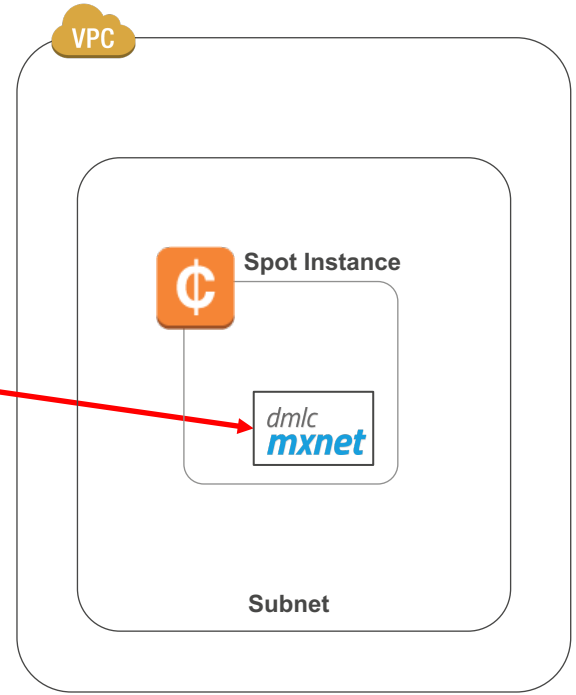
How does ECS use ECR?

dmlc mxnet

Amazon ECR

```
"containerDefinitions": [
 {
  "volumesFrom": [],
  "memory": 2048,
  "extraHosts": null,
  "dnsServers": null,
  "disableNetworking": null,
  "dnsSearchDomains": null,
  "portMappings": [
   {
    "hostPort": 80,
    "containerPort": 8888,
    "protocol": "tcp"
   }
  ],
  "hostname": null,
  "essential": true,
  "entryPoint": null,
  "mountPoints": [],
  "name": "mxnet",
  "ulimits": null,
  "dockerSecurityOptions": null,
  "environment": [],
  "links": null,
  "workingDirectory": null,
  "readonlyRootFilesystem": null,
  "image": "[redacted]dkr.ecr.us-west-2.amazonaws.com/test:latest",
```

Amazon ECS



AWS CloudFormation

CloudFormation – Components & Technology

Template

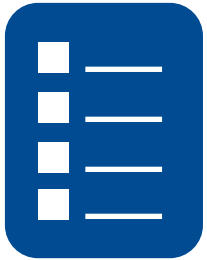
JSON or YAML formatted file

Parameter definition

Resource creation

Configuration actions

CloudFormation



Framework

Stack creation

Stack updates

Error detection and rollback

Stack

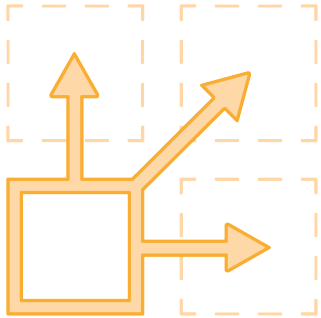
Configured AWS resources

Comprehensive service support

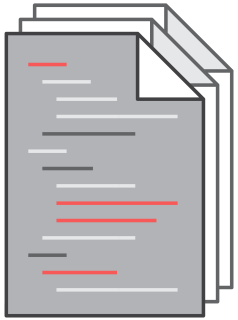
Service event aware

Customizable

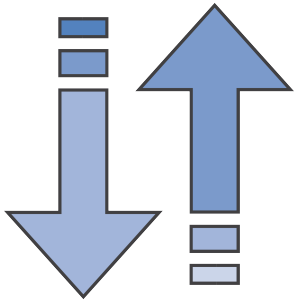
CloudFormation benefits



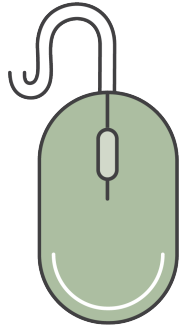
Templated resource provisioning



Infrastructure as code

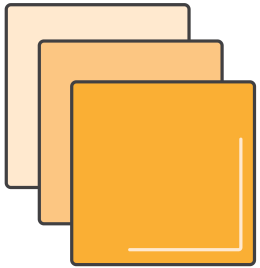


Declarative and flexible

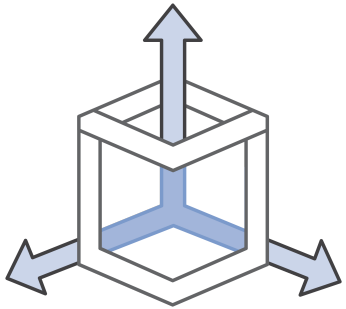


Easy to use

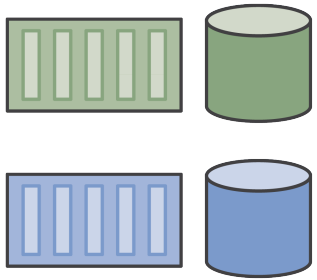
CloudFormation use cases



Stack replication



Infrastructure scale out



Blue-green deployments

Why do customers use CloudFormation?

Developers/DevOps teams value CloudFormation for its ability to treat infrastructure as code, allowing them to apply software engineering principles, such as SOA, revision control, code reviews, integration testing to infrastructure.

IT Admins and MSPs value CloudFormation as a platform to enable standardization, managed consumption, and role specialization.

ISVs value CloudFormation for its ability to support scaling out of multi-tenant SaaS products by quickly replicating or updating stacks. ISVs also value CloudFormation as a way to package and deploy their software in their customer accounts on AWS.

EC2 Spot Instances

Amazon EC2 consumption models

On-Demand

Pay for compute capacity by the hour with no long-term commitments

For spiky workloads, or to define needs

Reserved

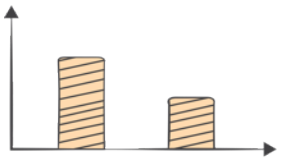
Make a low, one-time payment and receive a significant discount on the hourly charge

For committed utilization

Spot

Bid for unused capacity, charged at a Spot Price which fluctuates based on supply and demand

For time-insensitive or transient workloads



What are EC2 Spot Instances?

EC2 Spot Instances are

spare EC2 On-Demand capacity

with very simple rules…

With Spot the rules are simple

Markets where the price of compute changes based on supply and demand

You'll never pay more than your bid. When the market exceeds your bid you get 2 minutes to wrap up your work

Show me the markets!

C4	1a	1b	1c	On Demand
8XL	$0.50	$0.27	$0.29	$1.76
4XL	$0.21	$0.30	$0.16	$0.88
2XL	$0.08	$0.07	$0.08	$0.44
XL	$0.04	$0.05	$0.04	$.22
L	$0.01	$0.01	$0.04	$0.11

Each instance family

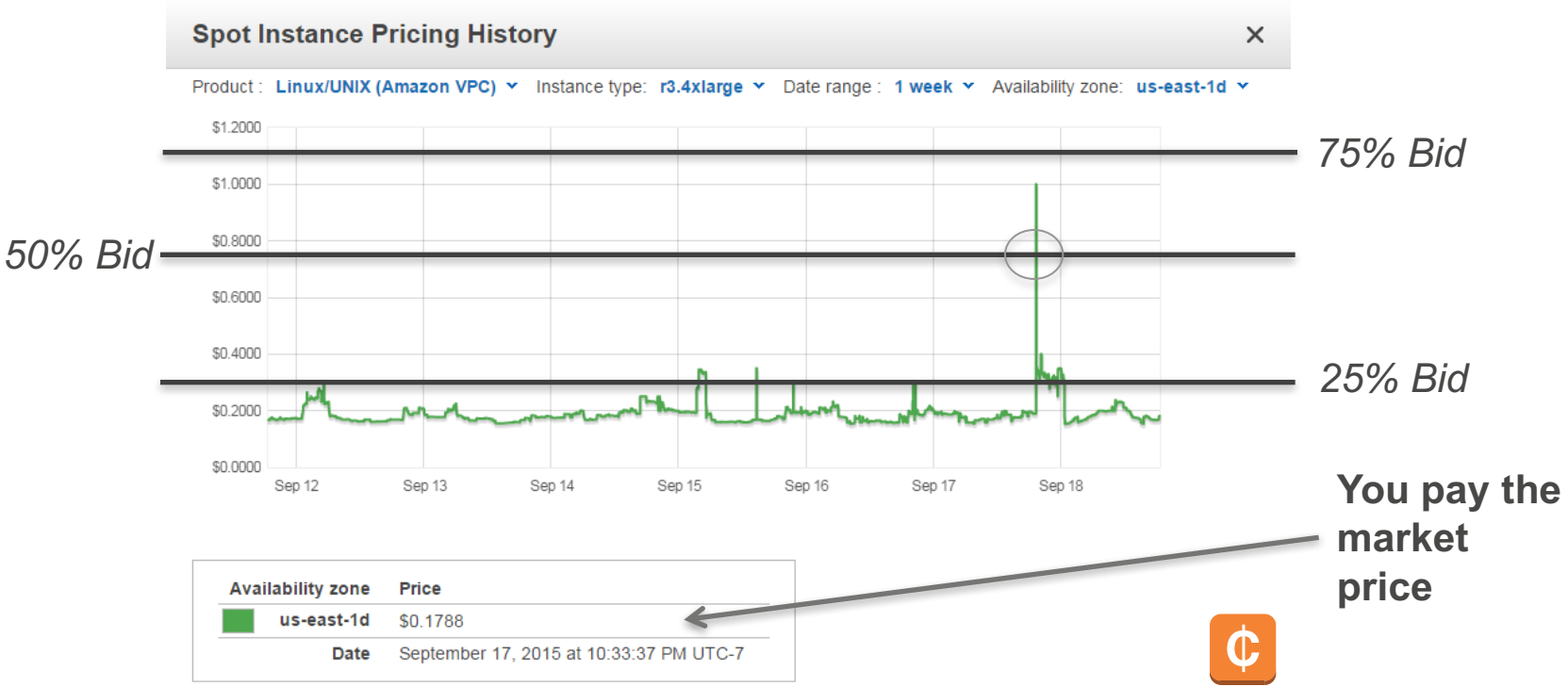
Each instance size

Each Availability Zone

In every Region

Is a separate ***Spot Market***

Bid Price vs Market Price



Why ECS and EC2 Spot Instances?

1. Get the best value for EC2 capacity

- Since Spot Instances typically cost 50-90% less than On-Demand, you can increase your compute capacity by 2-10x within the same budget

- Or you could save 50-90% on your existing workload

- Either way, you should try it!

2. Diversification with EC2 Spot Fleet

- Containers are a natural fit for a diverse allocation of resources – ECS just sees a pool of resources

- Spot fleet thrives on diversification- across instance types, instance sizes, and Availability Zones

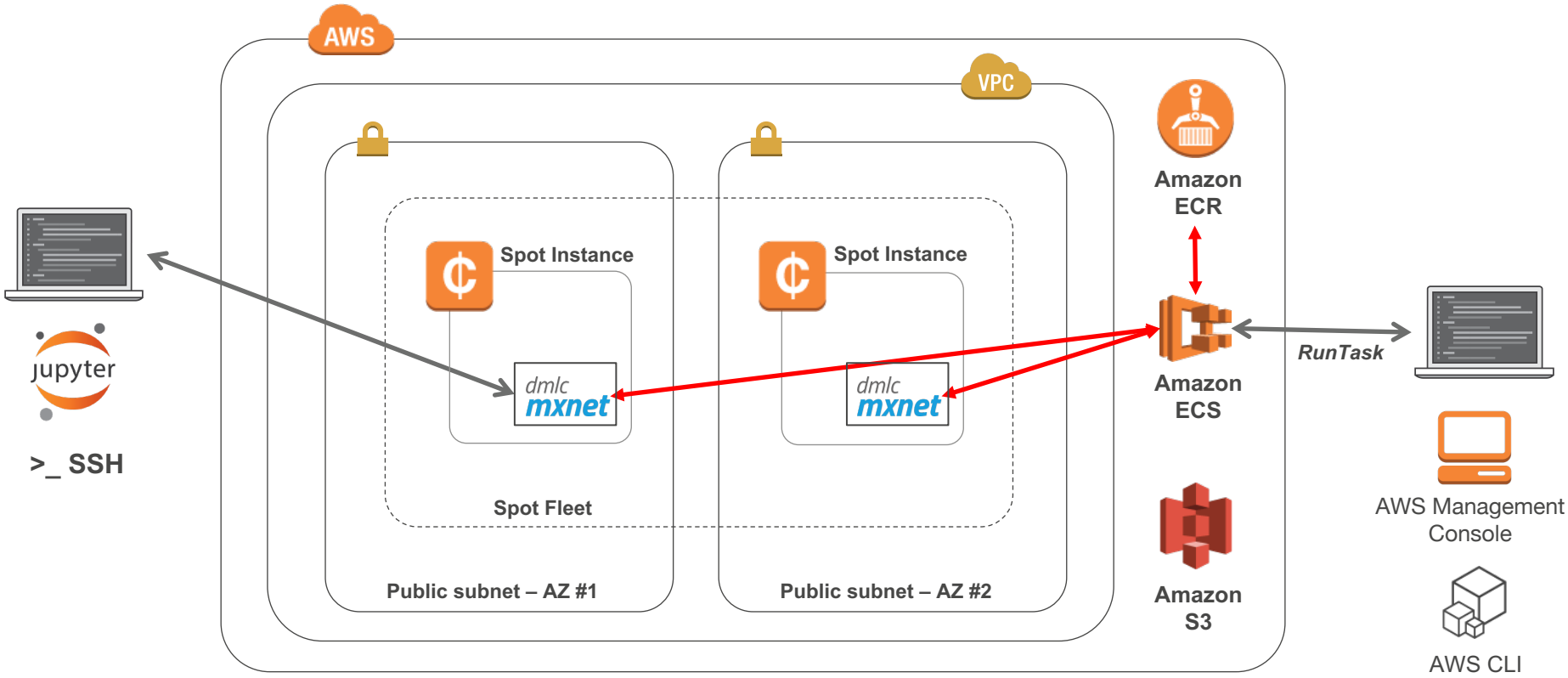
3. Lowest price with EC2 Spot Fleet

- The ECS runtask scheduler randomly distributes tasks across your cluster (typically used for batch jobs)
- Spot Fleet has a built-in allocation strategy of Lowest Price
- Also don’t forget about Spot Blocks

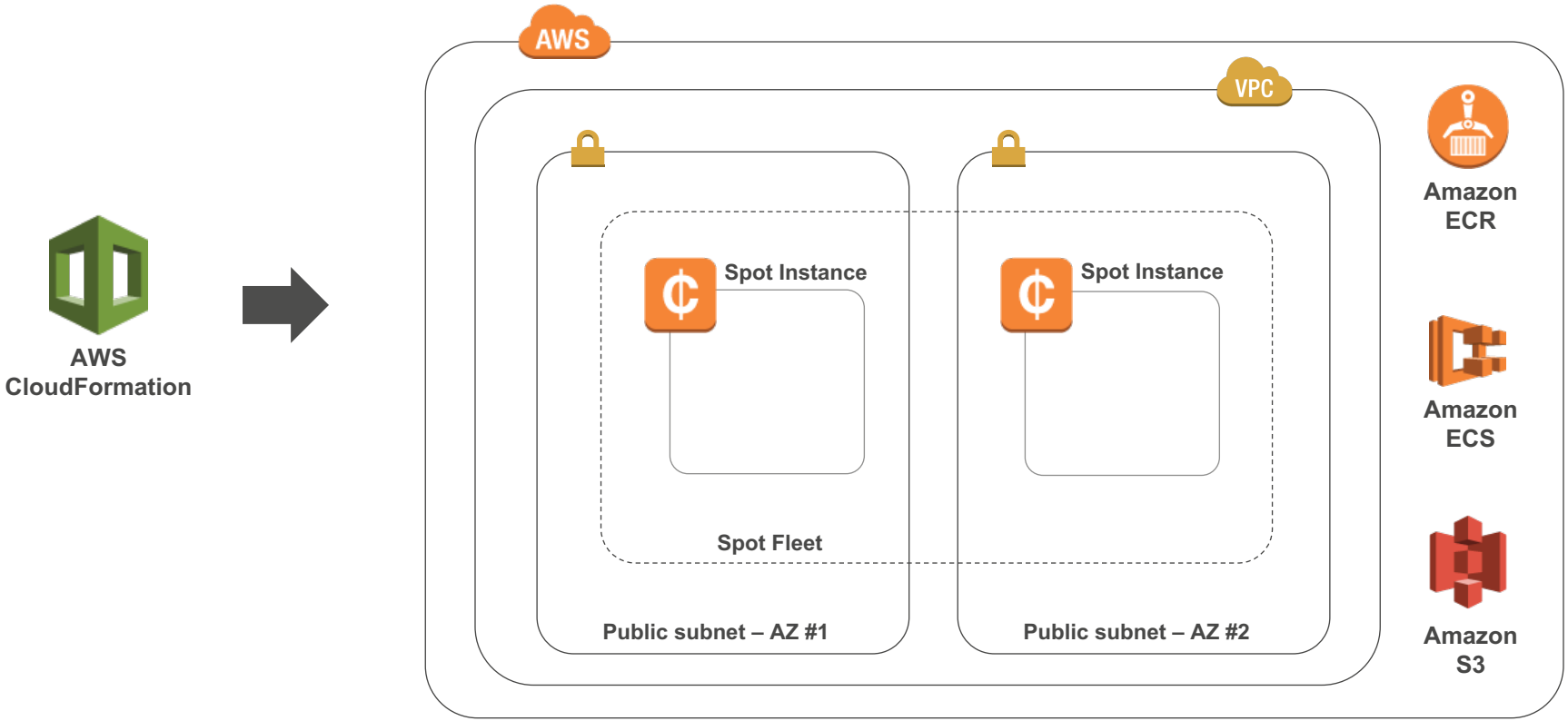
  (run Spot Instances without interruption for 1 to 6 hours)

Workshop: Image Classification

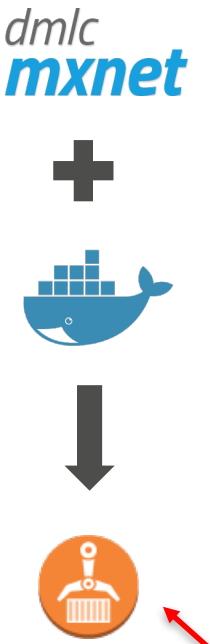
Overall architecture



Lab 1: Set Up the Workshop Environment

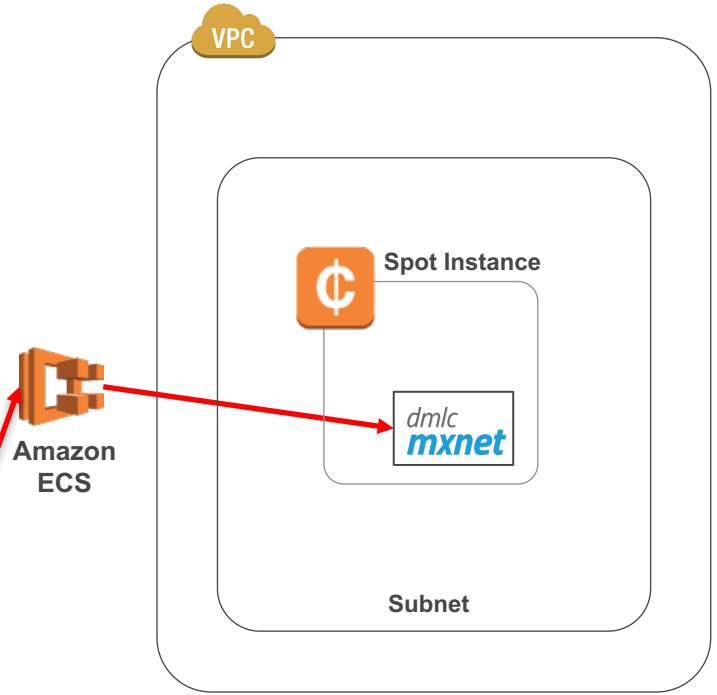


Lab 2: Build an MXNet Docker Image

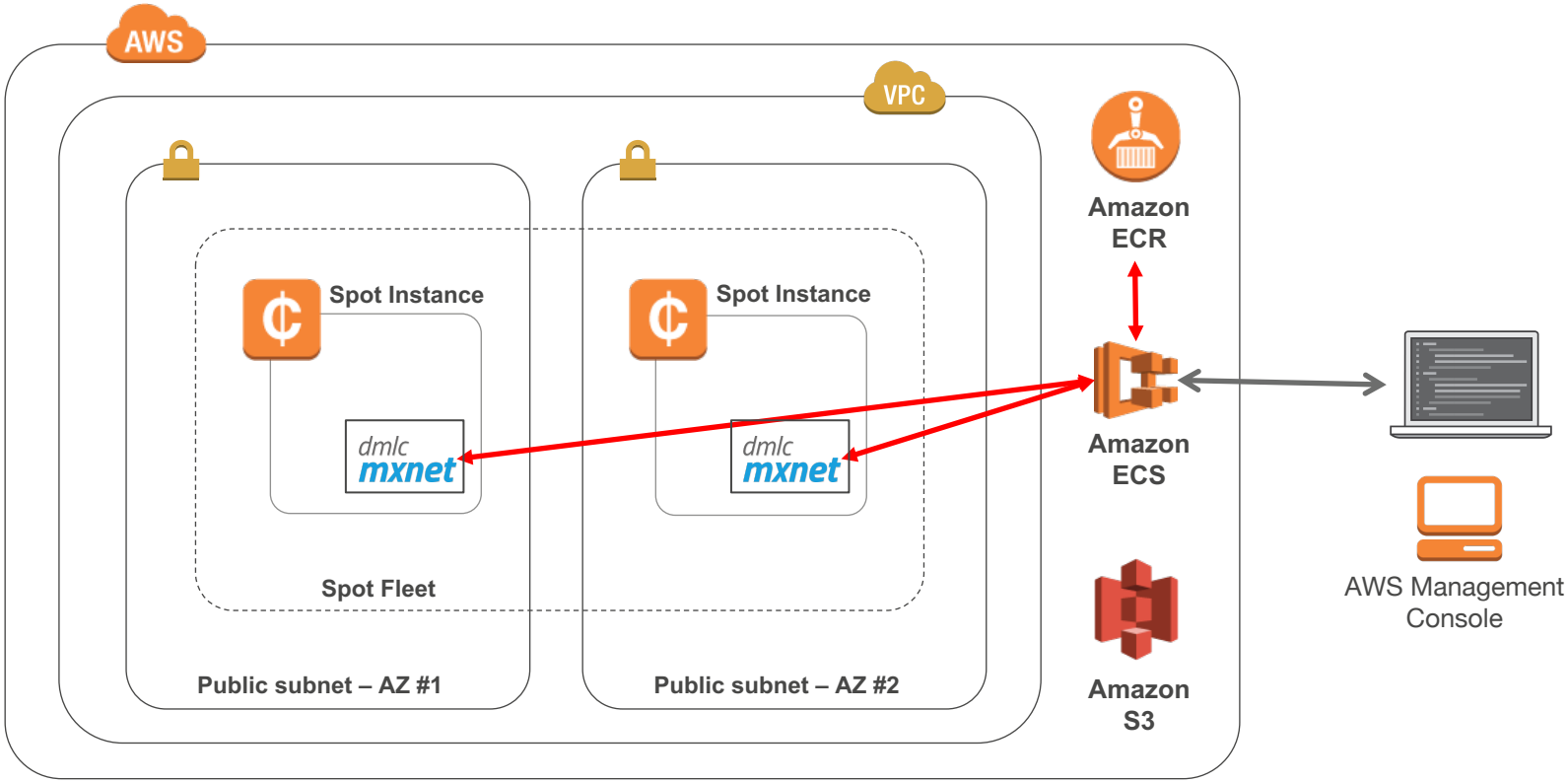



```
"containerDefinitions": [
 {
  "volumesFrom": [],
  "memory": 2048,
  "extraHosts": null,
  "dnsServers": null,
  "disableNetworking": null,
  "dnsSearchDomains": null,
  "portMappings": [
   {
    "hostPort": 80,
    "containerPort": 8888,
    "protocol": "tcp"
   }
  ],
  "hostname": null,
  "essential": true,
  "entryPoint": null,
  "mountPoints": [],
  "name": "mxnet",
  "ulimits": null,
  "dockerSecurityOptions": null,
  "environment": [],
  "links": null,
  "workingDirectory": null,
  "readonlyRootFilesystem": null,
  "image": "[redacted]dkr.ecr.us-west-2.amazonaws.com/test:latest",
```

ECS Task Definition



Lab 3: Deploy MXNet Container with ECS

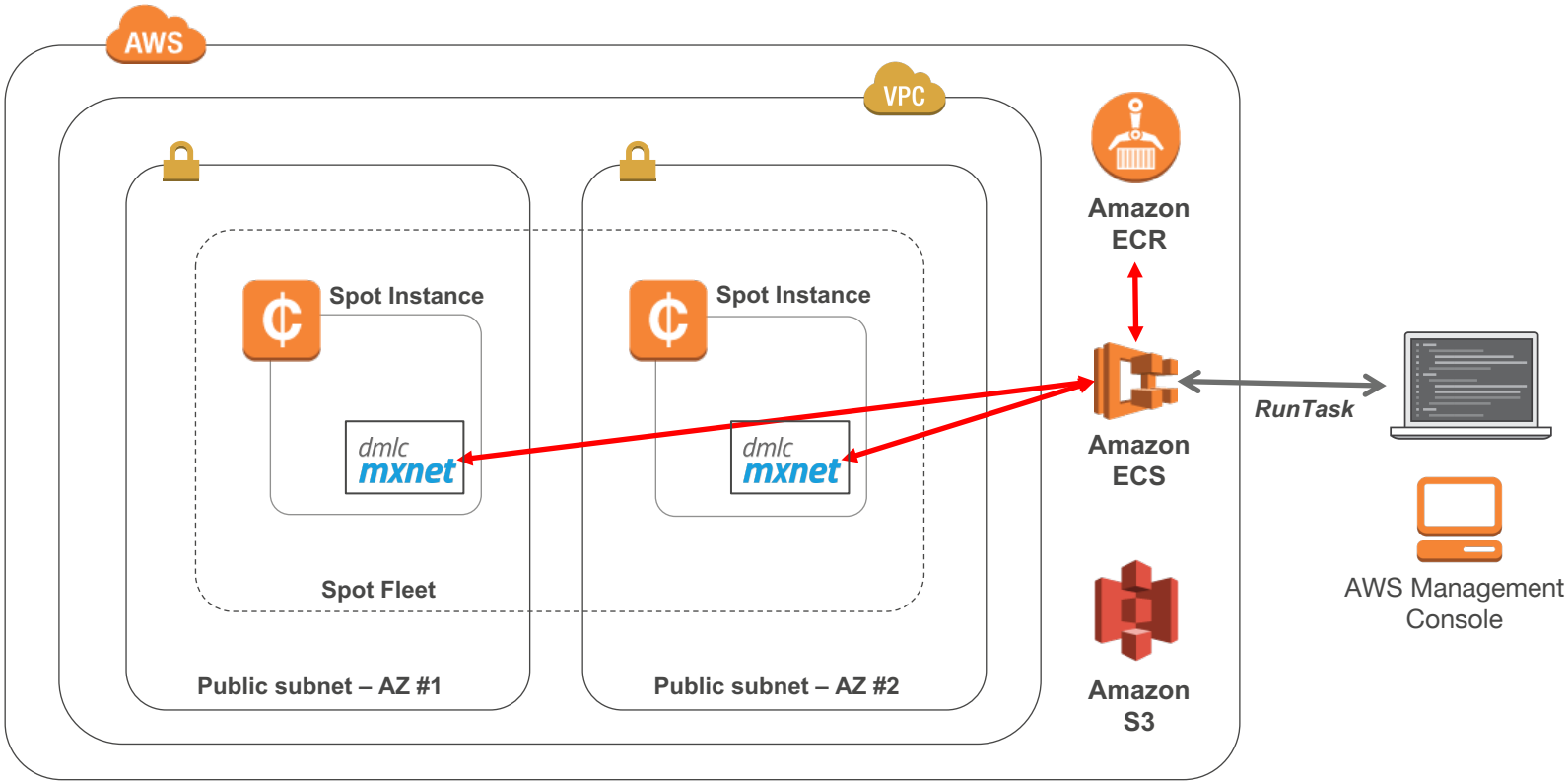


Lab 4: Image Classification Demo

```
In [4]: url = 'http://writm.com/wp-content/uploads/2016/08/Cat-hd-wallpapers.jpg'
        predict(get_image(url), mod, synsets)
```

```
probability=0.692329, class=n02122948 kitten, kitty
probability=0.043847, class=n01323155 kit
probability=0.030002, class=n01318894 pet
probability=0.029693, class=n02122878 tabby, queen
probability=0.026972, class=n01322221 baby
```

Lab 5: Wrap Image Classification in an ECS Task



Some pointers…

- Apply your AWS Credits.
  - https://aws.amazon.com/awscredits/
- This is a self-paced lab. Don’t stress to finish here. You can access the content from GitHub anytime.
- The value-add of doing the workshop here is being together. Don’t be shy. Ask questions.
- Work together. Collaborate. Most importantly, **have fun!**

Let’s get started!

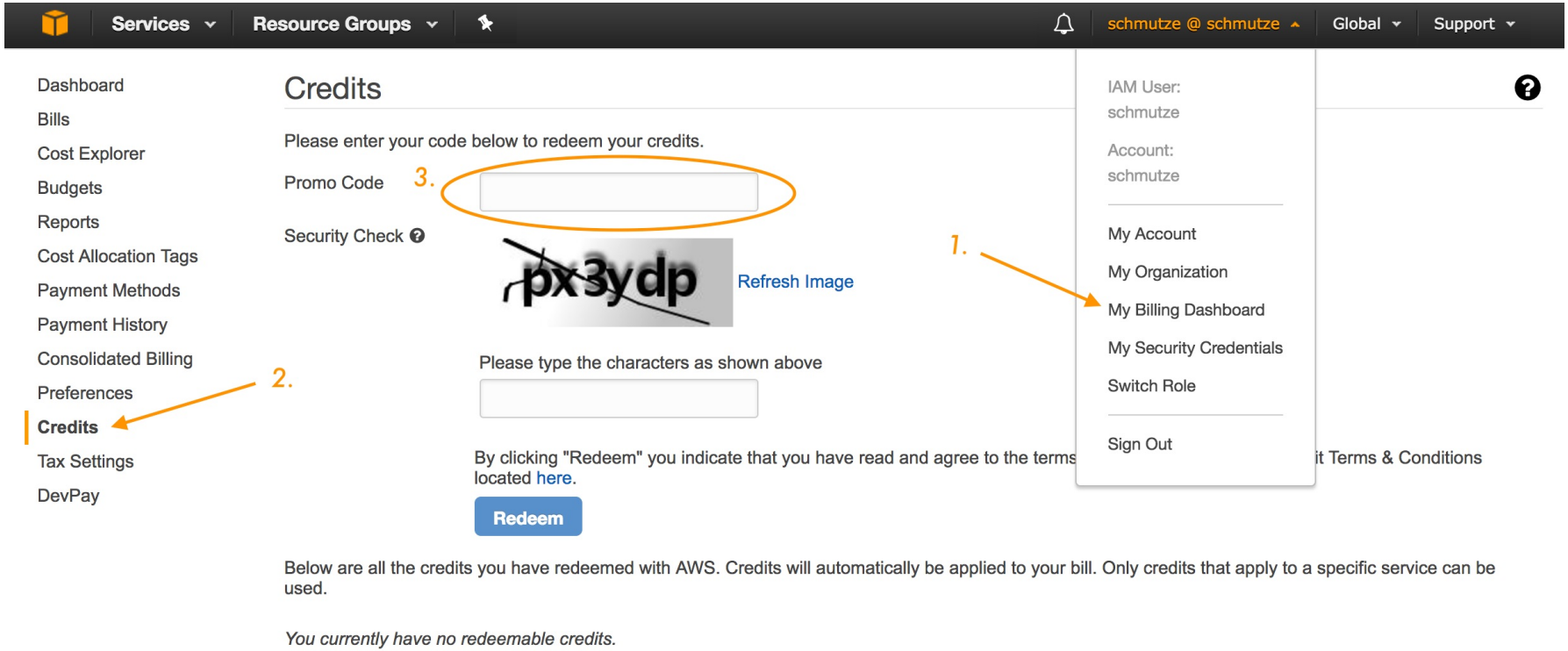
https://github.com/awslabs/ecs-deep-learning-workshop

Follow the lab guide! Raise your hand if you have questions.

Appendix

Apply Your Credits

https://aws.amazon.com/awscredits/



Workshop Cleanup

1. Delete the CloudFormation stack
2. Check what resources CloudFormation was not able to delete (it won't delete things it did not create or that were modified)
    1. S3 Bucket
    2. ECR Repository
3. Delete the CloudFormation stack again

Evaluations

Please don’t forget to complete the workshop evaluation in the app!

Related Sessions

- **Deep Dive into Apache MXNet on AWS** (BDA401)
- **Getting the most Bang for your buck with #EC2 #Winning** (SRV301)
- **Deep Learning at Cloud Scale and AI as a Service** (DEM307)

AWS

SUMMIT

Thank You!

amazon web services