CMP316

AWS re:INVENT

Hedge Your Own Funds: Run Monte Carlo Simulations on EC2 Spot Fleet

Shawn O'Connor – Solutions Architect

November 28, 2017

AWS re:Invent

aws

Before we get started

- You will be using your own AWS account in this workshop
- If your AWS account is <24 hours old, or you have never launched an Amazon EC2 instance in your account, please do one of the following:
    1. Raise your hand and provide your AWS account # to one of us
    2. Email your AWS Account # to shawo@amazon.com

What to expect from this workshop

- Introduction to the workshop
- Overview of AWS CloudFormation
- Overview of Amazon EC2 Spot instances
- Overview of AWS Batch
- Hands on, self-paced workshop
- The goal is to learn about EC2 Spot instances and batch processing, not become an expert in algorithmic trading

Workshop introduction/architecture

AWS re:Invent

aws

What is algorithmic trading?

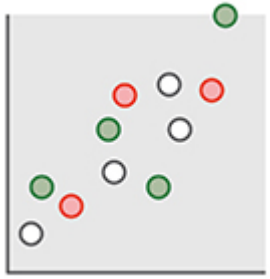
Algorithmic trading, also referred to as Algo trading and black box trading, is a trading system that utilizes advanced and complex mathematical models and formulas to make high-speed decisions and transactions in the financial markets.

Source: Investopedia.com

Algorithmic trading pipeline

Data analysis

Explore Ideas to define a trading strategy



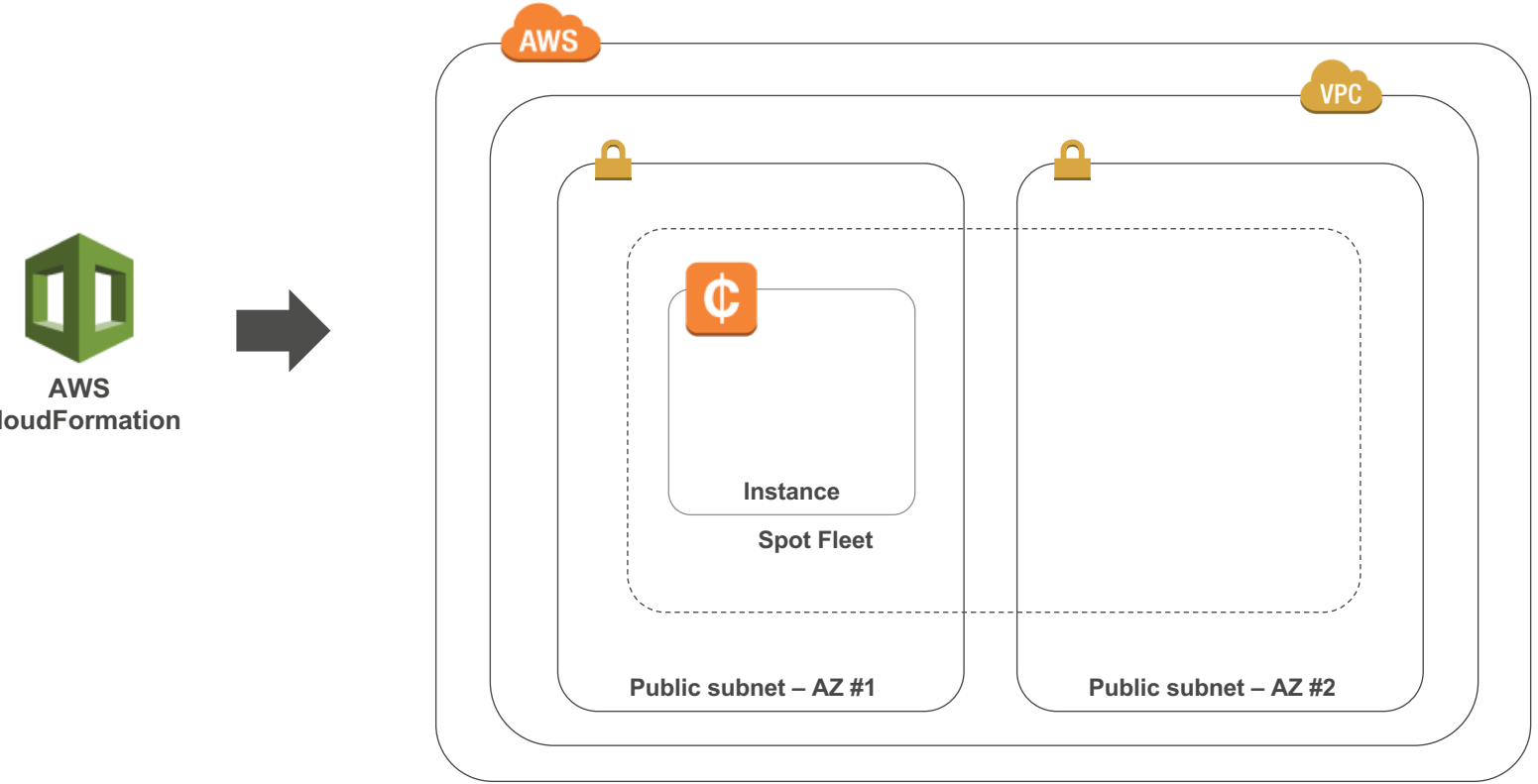
Backtesting

Test out our strategy on historical data

Forward testing

Simulate trades with real-time data

Lab 1–Deploy the workshop environment



Lab 2–Trading strategy demo with Jupyter

File Edit View Insert Cell Kernel Help

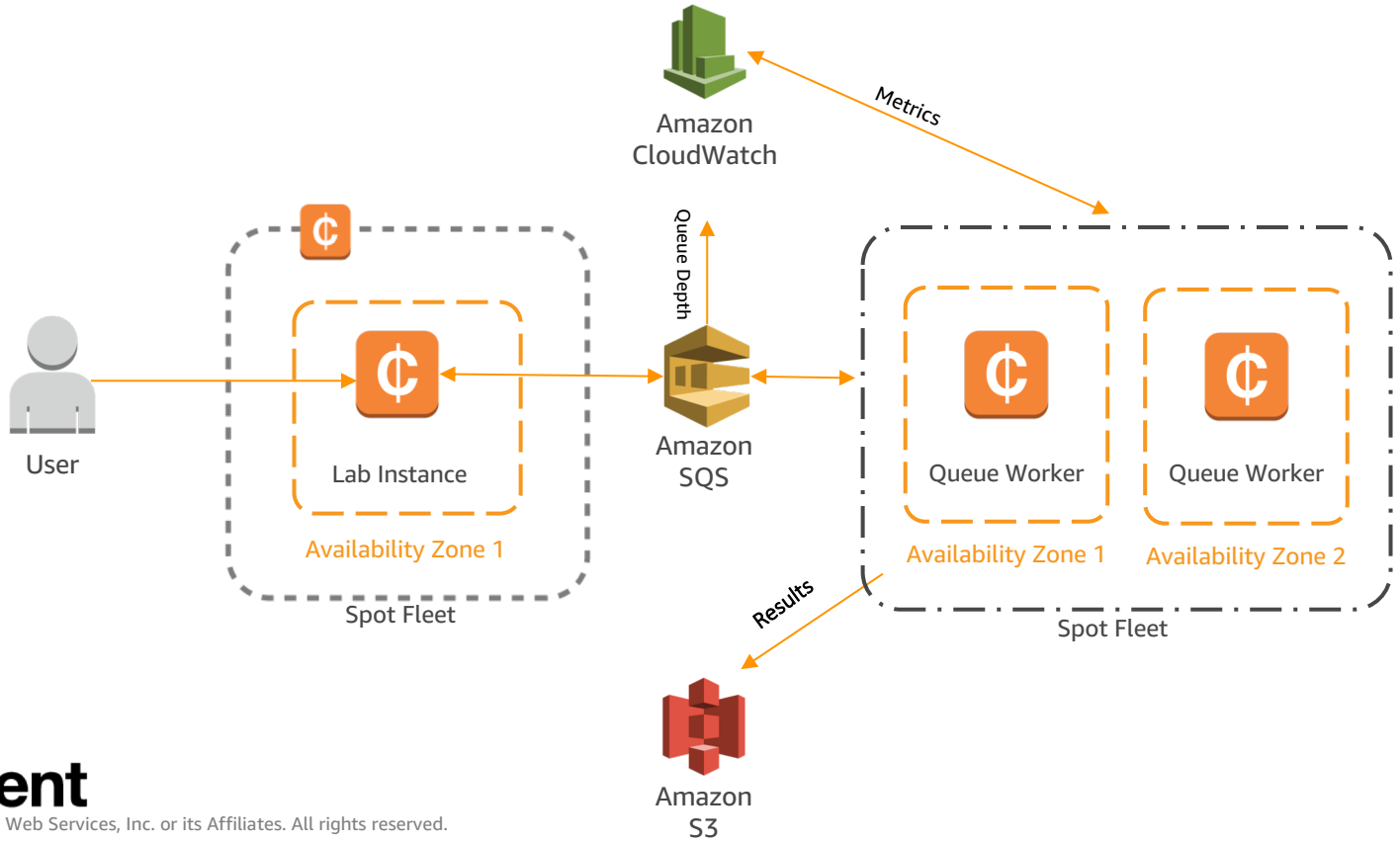
Code

Date						
2006-10-02	31.980000	32.029999	30.830000	30.870001	30.870001	6815000
2006-10-03	30.900000	32.000000	30.580000	31.700001	31.700001	8034000
2006-10-04	31.750000	32.830002	31.299999	32.759998	32.759998	7064800
2006-10-05	32.680000	33.400002	32.400002	33.320000	33.320000	8455900
2006-10-06	33.150002	33.220001	32.500000	32.590000	32.590000	4219200

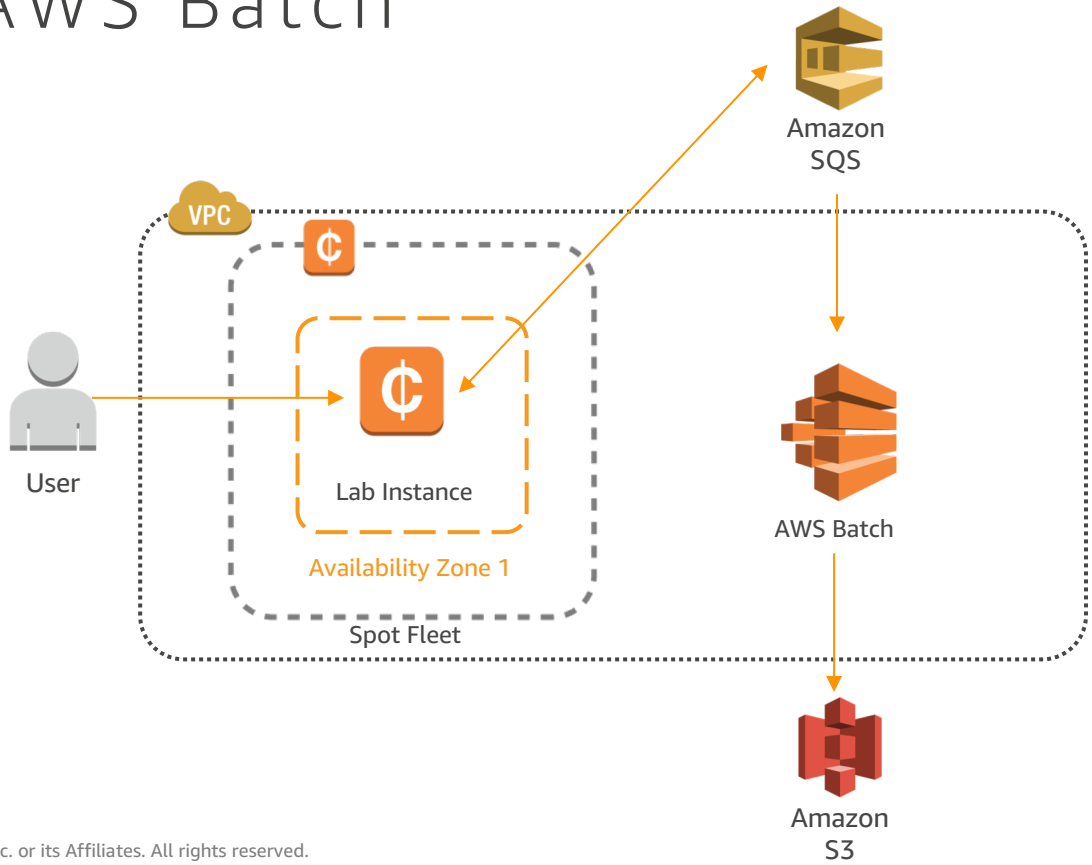
In [6]:

```
# plot the closing price
stock_df['Close'].plot()
plt.show()
```

Lab 3–Deploy an automated trading strategy



Lab 4 - Leverage a Fully Managed Solution using AWS Batch



AWS CloudFormation

AWS CloudFormation components & technology

Template

JSON or YAML file

Parameter definition

Resource creation

Configuration actions

AWS CloudFormation

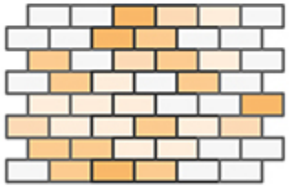
Framework

Stack creation

Stack updates

Error detection and rollback

Stack



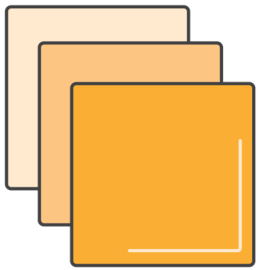
Configured AWS resources

AWS service support

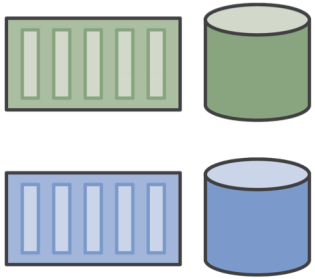
Service event aware

Customizable

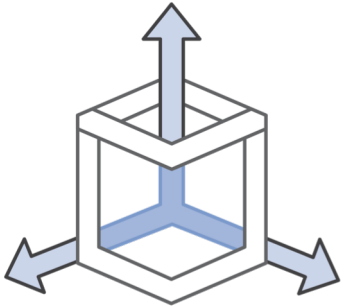
CloudFormation use cases



Stack replication



Blue-green deployments



Infrastructure scale out

Multi-region deployments

Why customers use AWS CloudFormation

Developers/DevOps teams value AWS 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 AWS CloudFormation as a platform to enable standardization, managed consumption, and role-specialization.

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

Amazon EC2 Spot

AWS re:Invent

aws

AWS consumption models

On-demand

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

For spiky workloads, or to define needs

Reserved

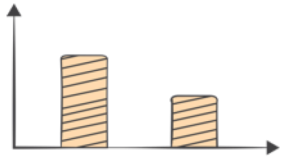
Make a low, one-time payment and receive a significant discount on the hourly charge

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 two minutes to wrap up your work.

Show me the markets!

C3	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

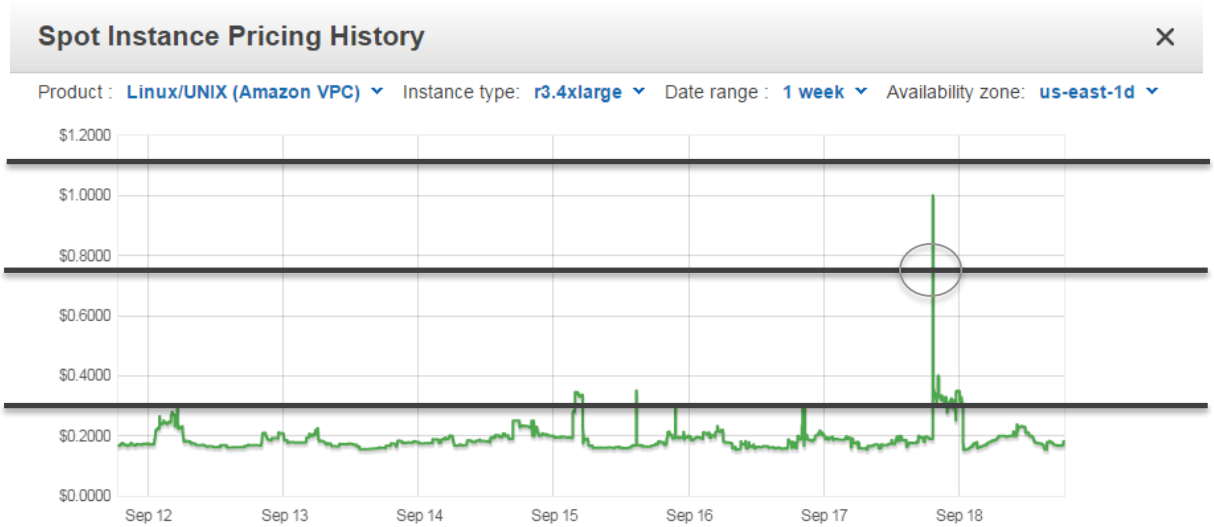
Each instance size

Each Availability Zone

In every region

Is a separate Spot Market

Bid price as opposed to market price



75% Bid

50% Bid

25% Bid

You pay the market price

EC2 best practices

Stateless

Fault tolerance

Multi-AZ

Loosely coupled

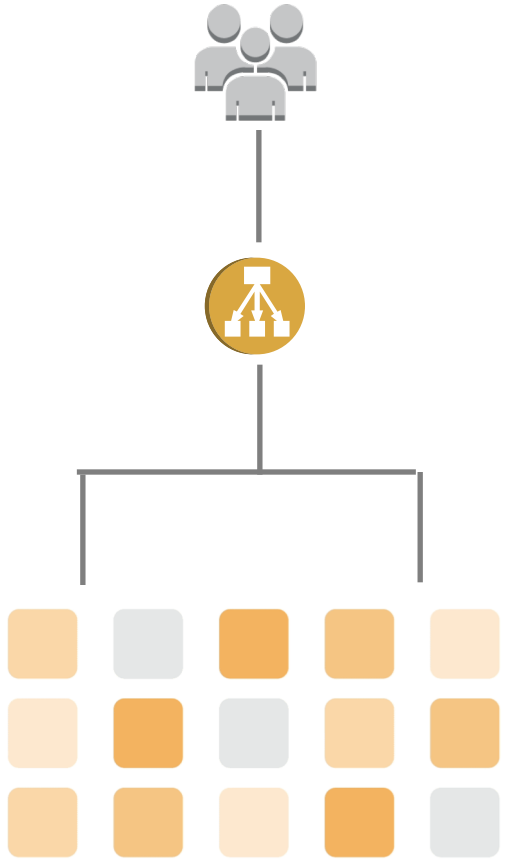
Instance flexibility

Diversify with EC2 Spot Fleets

Multiple EC2 Spot instances selected

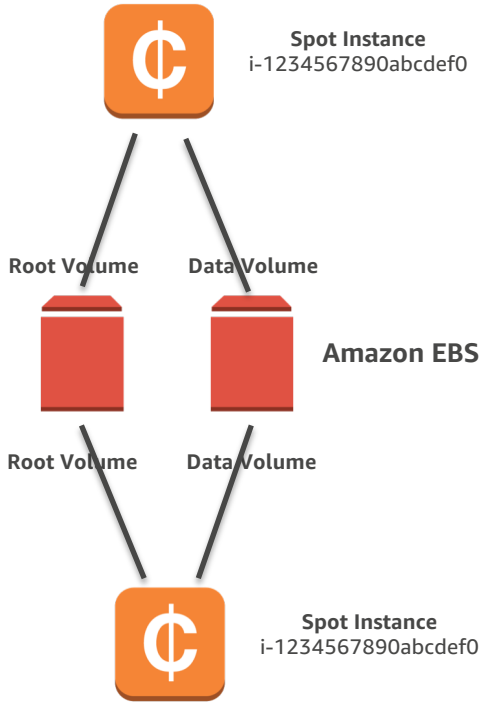
Multiple Availability Zones selected

Pick the instances with similar performance characteristics e.g., c3.large, m3.large, m4.large, r3.large, c4.large.



Persist workloads with EC2 stop/start

- Amazon EC2 Spot now allows EBS-backed instances to be stopped, instead of being terminated
- The EBS root device and attached EBS volumes are saved, and their data persists
- Upon restart, the EBS root device is restored from its prior state and the instance retains its instance ID



Get the best value from EC2

- Since Spot instances typically cost 50-90% less than on-demand, you can **increase your compute capacity by 2-10x within the same budget**

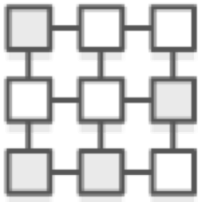
- Or you could **save 50-90% on your existing workload**

- Either way, **you should try it!**

AWS Batch

What is batch computing?

Run jobs asynchronously and automatically across one or more computers.



Jobs may have dependencies, making the sequencing and scheduling of multiple jobs complex and challenging.

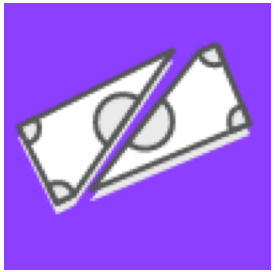
Introducing AWS Batch

Fully managed

No software to install or servers to manage; AWS Batch provisions, manages, and scales your infrastructure

Integrated with AWS

Natively integrated with AWS, AWS Batch jobs can easily and securely interact with services such as Amazon S3, Amazon DynamoDB, and Amazon Rekognition



Cost-optimized

AWS Batch automatically provisions compute resources tailored to the needs of your jobs using Amazon EC2 and EC2 Spot

AWS Batch concepts

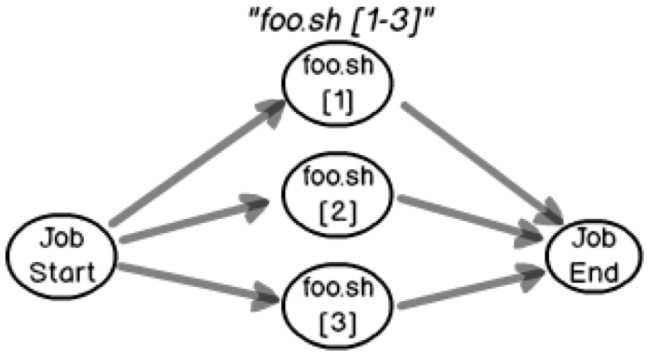
Jobs - Unit of work executed by AWS Batch as containerized applications running on Amazon EC2.
Job Queue - Managed queues where jobs reside until they are able to be scheduled to a compute resource.
Compute Environments - EC2 used to run containerized batch jobs. You can specify Spot as a % of on-demand.
Scheduler - Evaluates when, where, and how to run jobs that have been submitted to a job queue.

Easily run massively parallel jobs

Today, users can submit a large number of independent "**simple jobs**." Soon, we will add support for "**array jobs**" that run many copies of an application against an array of elements.

Array jobs are an efficient way to run:

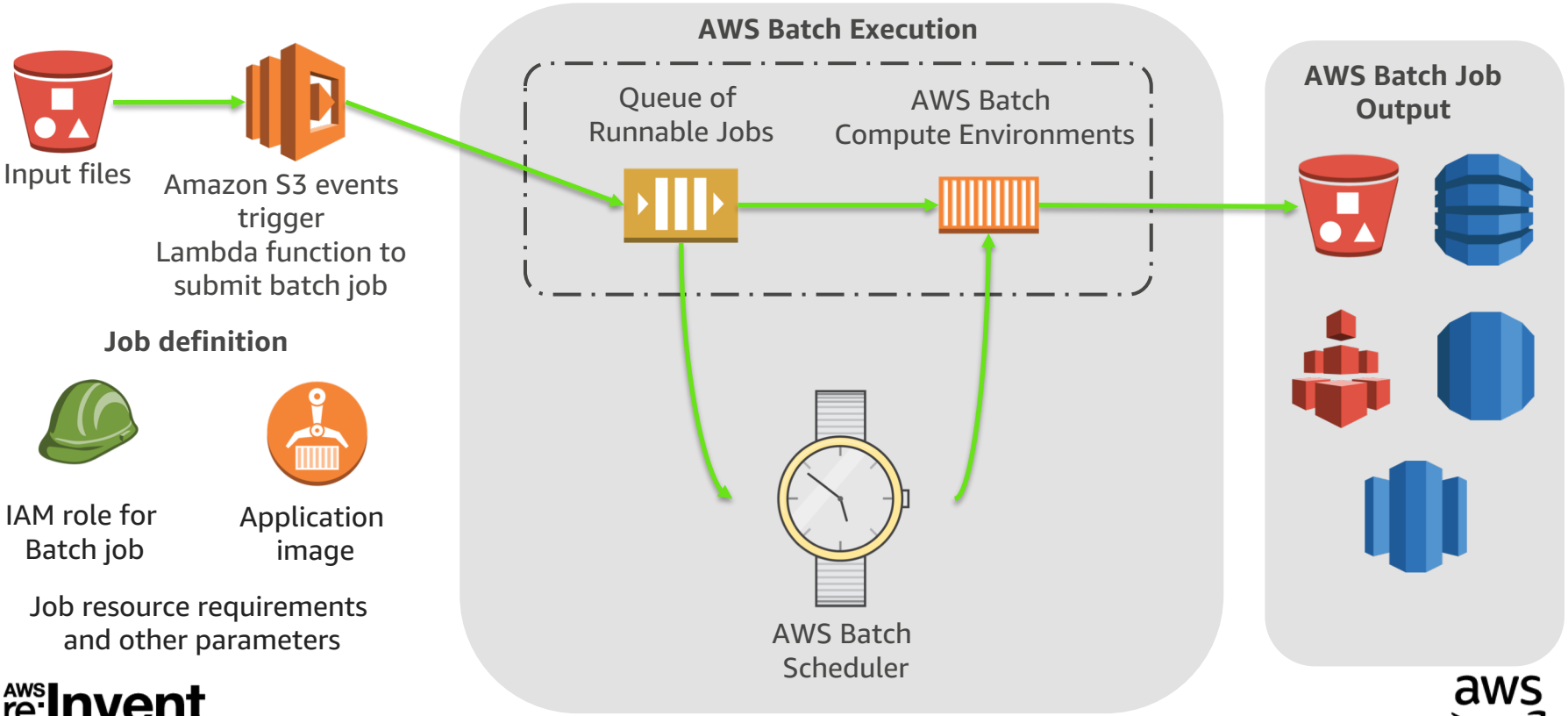
- Parametric sweeps
- Monte Carlo simulations
- Processing a large collection of objects



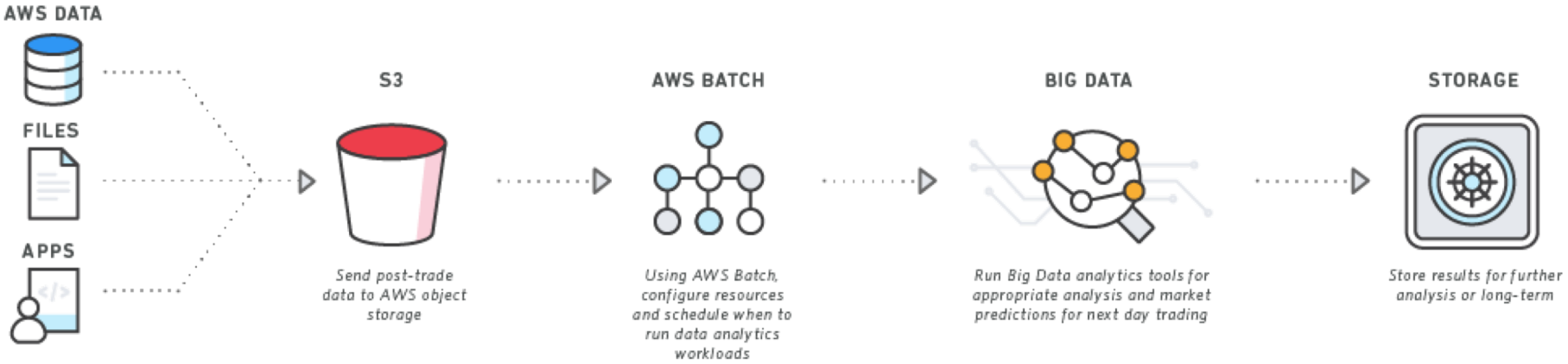
But, how much does it cost?

There is no charge for AWS Batch; you only pay for the underlying resources that you consume!

Typical AWS Batch job architecture



Financial trading analytics



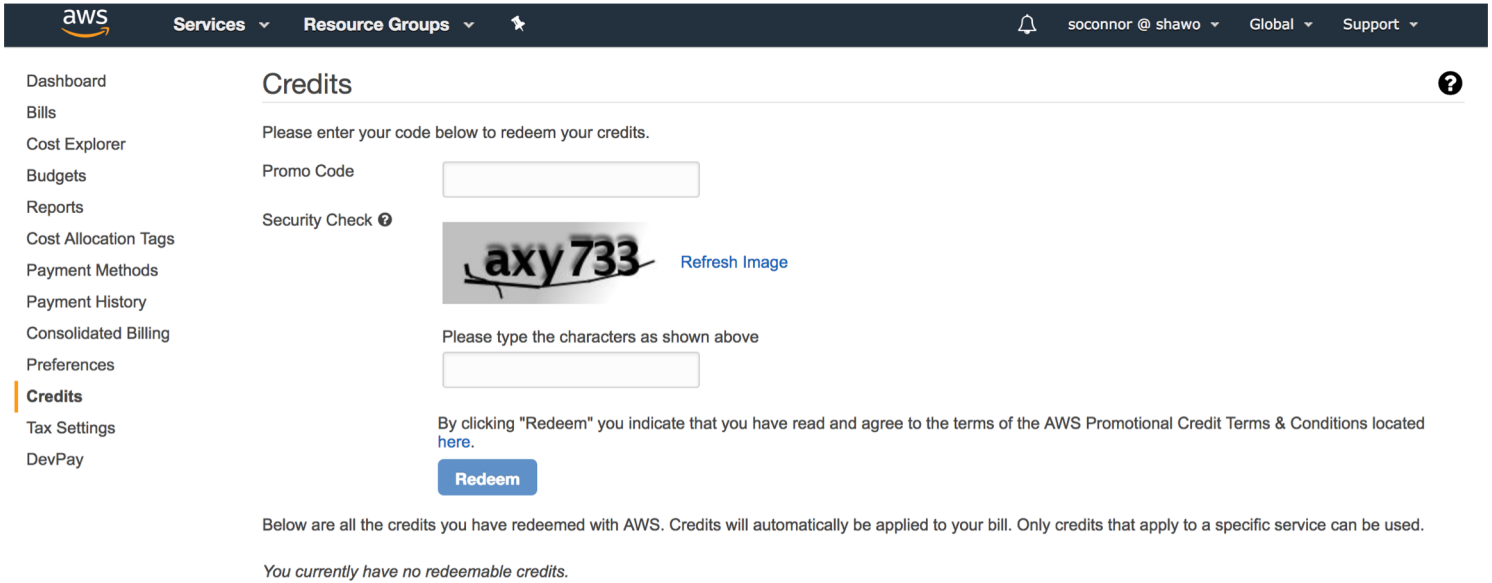
After the workshop

Some pointers...

- Apply your AWS Credits.
- This is a self-paced lab. Don’t stress to finish here. You can access the content from GitHub anytime.
- The value of doing the workshop here is being together. Ask questions.
- Experiment and expand on the examples given
- Work together. Collaborate. Most importantly, **have fun!**

Apply your credits

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



Workshop cleanup

1. Empty the S3 bucket(s)
2. Delete manually created resources
    1. SQS Queue
    2. AWS Batch Components
    3. Spot Fleet request for workers
3. Delete the AWS CloudFormation stack
4. Check what resources AWS CloudFormation was not able to delete (it won't delete things it did not create or that were modified)
5. Delete the AWS CloudFormation stack again

Evaluations

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

AWS re:Invent

aws