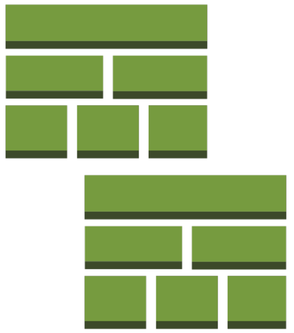
Developing and Running Applications in AWS Batch

Noora Siddiqui
October 2019

A high level overview of running scalable workloads in AWS, which includes a brief description of the infrastructure components involved and a demonstration of how to provision and connect resources into a batch processing environment.
DEMO: Developing an application that works with either local filesystem paths or S3 URIs. This includes payload code, Dockerfile, building, deployment, and operation.

AWS CloudFormation

Easy way to “template” the creation of related AWS resources and predictably provision them.



Resource Stacks

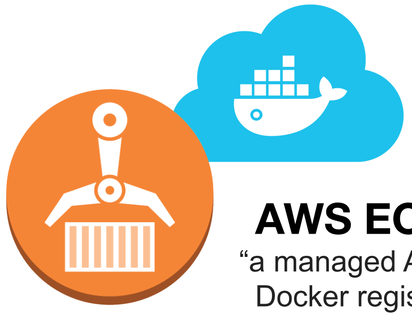
Includes the Virtual Private Cloud (VPC), its associated subnets, and the compute environments specifying instance types and billing constructs

AWS Batch

“plans, schedules, and executes your **batch** computing workloads”

BATCH LAYER*

Where data processing occurs



AWS ECR

“a managed AWS Docker registry service”

AWS S3

Storage service; simply, the easiest piece to understand here

JOB LAYER*

A reservoir for 1. definitions supporting each analytical application and 2. the input and transformed data

AWS EC2

scalable computing capacity (servers, security, networking, storage)

AWS IAM

Controls permissions and access to all AWS services and resources

WORKFLOW LAYER*

State-monitoring machinery & orchestration to manage job dependencies & trigger job submission.

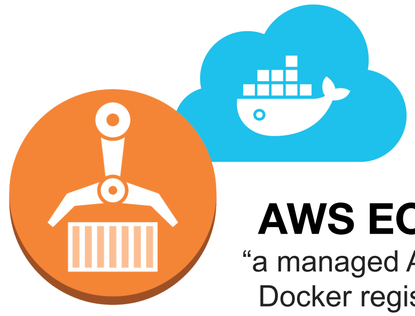
AWS StepFunctions

AWS Lambda

AWS CloudWatch

AWS Batch

“plans, schedules, and executes your **batch** computing workloads”



AWS ECR

“a managed AWS Docker registry service”

AWS S3

Storage service; simply, the easiest piece to understand here

JOB LAYER*

A reservoir for
1. definitions supporting each analytical application and
2. the input and transformed data

Components of AWS Batch

Job
Same idea as an HPC job; a basic unit of work.

Job Queue
Synonymous with the idea of queues on a cluster. However, these queues are relatively simple to make and edit. You can assign priority values to them easily and connect or detach them from compute environments, as needed.

Compute Environment
The compute resources (instance type, billing construct, IAM roles) associated with a queue. I'm using managed environments so AWS Batch handles scaling.

Job Definition
Unique to AWS Batch, includes some of the pieces found in HPC PBS directives. The definition specifies parameters that will be supplied to the job (e.g. number of vCPUs and memory), as well as the **docker image**, environmental variables, and the basic command that will be supplied at run time. Certain pieces of the job definition can be overridden at runtime.

Quick Look: AWS S3

In AWS S3, “buckets" and “objects” are the primary resources, with objects stored in buckets. Unlike a file system, S3 has a “flat structure.”

Let’s look at S3URIs:

[1] → aws s3 ls s3://bucket/key
[2] → aws s3 ls s3://drageneval/test/somefile.ext
[3] → aws s3 ls s3://drageneval/test

The best S3 equivalent I could find for the UNIX “tree” command:

[4] → aws s3 ls --summarize --human-readable --recursive s3://drageneval/test/

To Run an Application at Batch Scale in AWS

1. Know your application and all of the upstream (granular) dependencies
   EX: Alignstats <— HTSlib1.9 <— gcc libbz2-dev liblzma-dev libncurses5-dev libncursesw5-dev zlib1g-dev

2. Develop a wrapper script to take your application command and transform it under the hood to run locally to a Batch instance and perform the required I/O to/from S3 in a secure and time-efficient manner.
   EX: With the job command “alignstats -i <S3_INPUT> -o <S3_OUTPUT> -m <S3_INPUT> -C” AWS Batch should handle secure transfer of input, temporary output, and transfer of output to the expected location in S3.

3. Containerize 1 and 2 a lightweight docker image using a trusted upstream build

4. Set up an Amazon Machine Image (AMI) optimized for Batch and your application with the appropriate encrypted volumes — map these so that the instance host volume will mount to container in the way you expect.

5. Set up the compute environment(s) and job queue(s)

6. Set up the job definition to orchestrate 3, 4, 5 and 6

Wrapper Script, Dockerfile, S3 I/O

Demo