Pipleline for user review translation and sentiment analysis

Software Requirements Specifications

SRS can be found here

Solution Presented

The solution presented fulfils the requirement of deploying a Python-based program on AWS, provided that the architecture is cost-efficient, time-constrained and follows best practices.

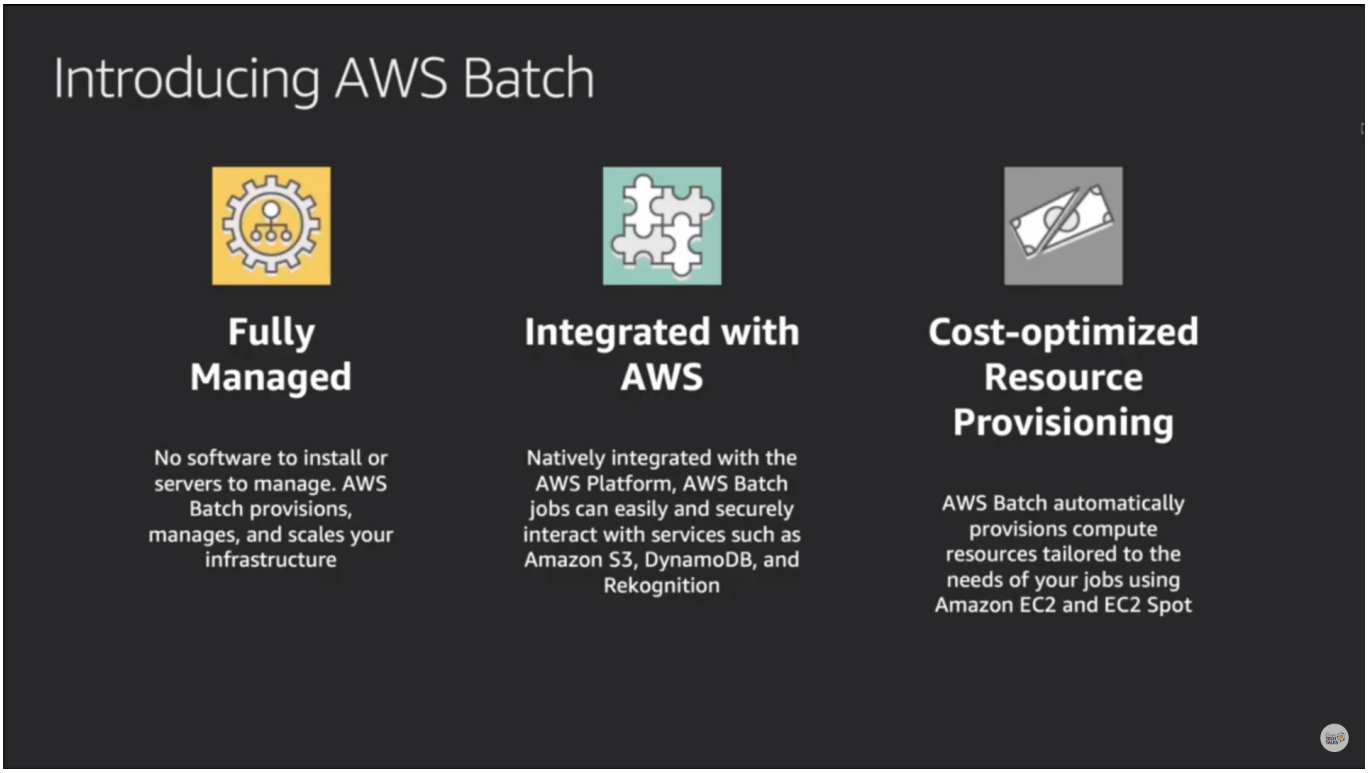
AWS

AWS Batch + Spot Instances

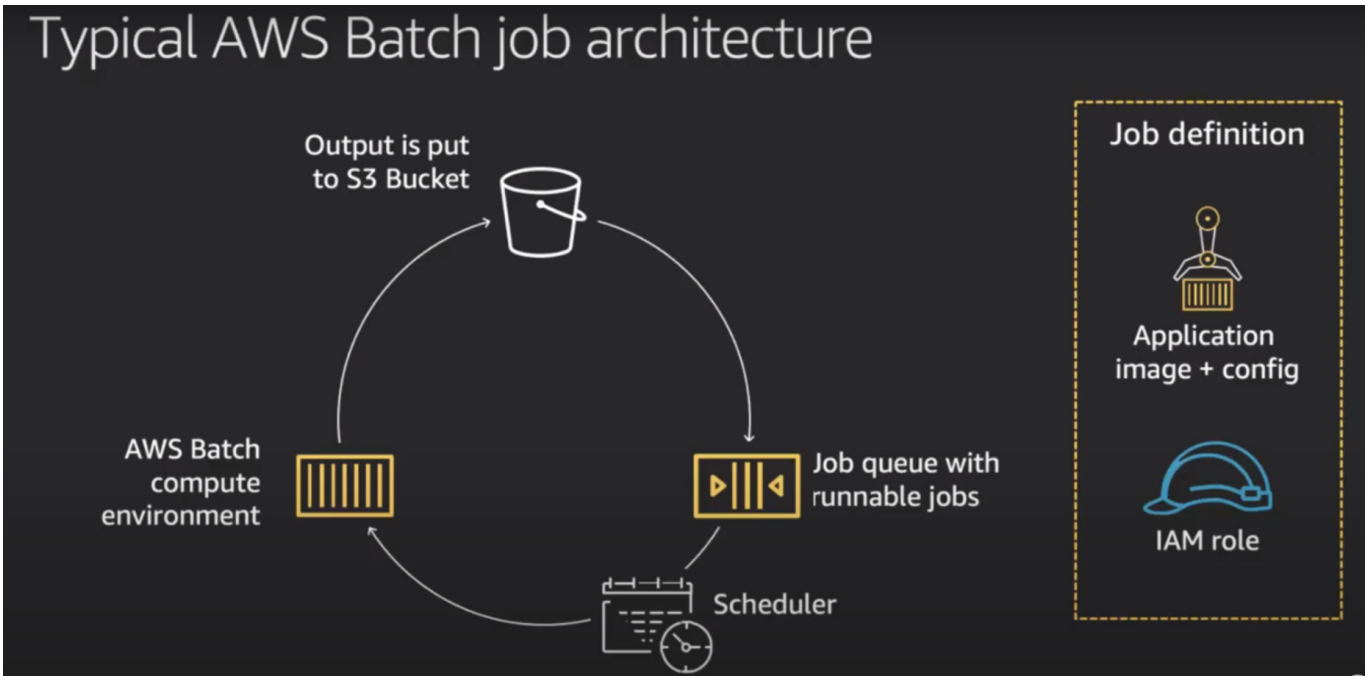
Note: All images below are screenshots from these

- Reference
- Reference

What is AWS Batch?

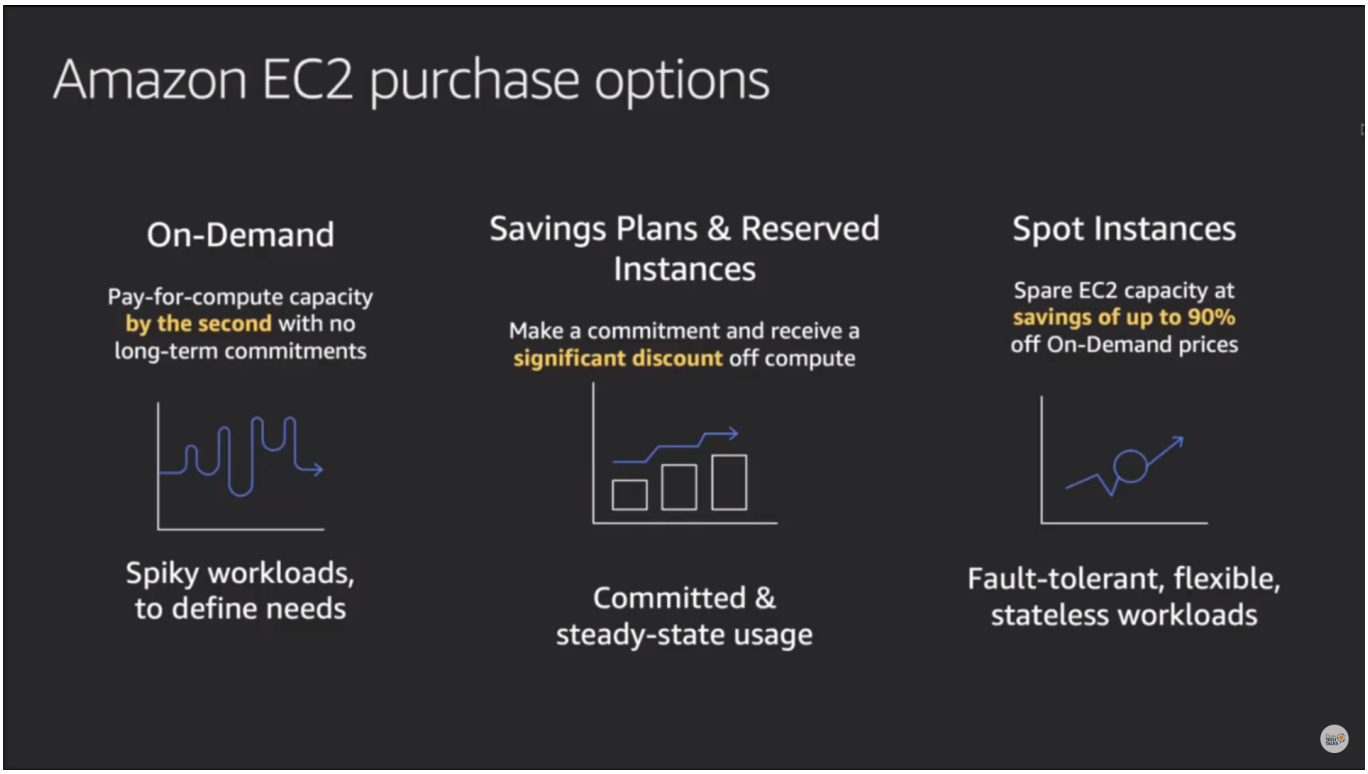


2. AWS Batch Job Architecture

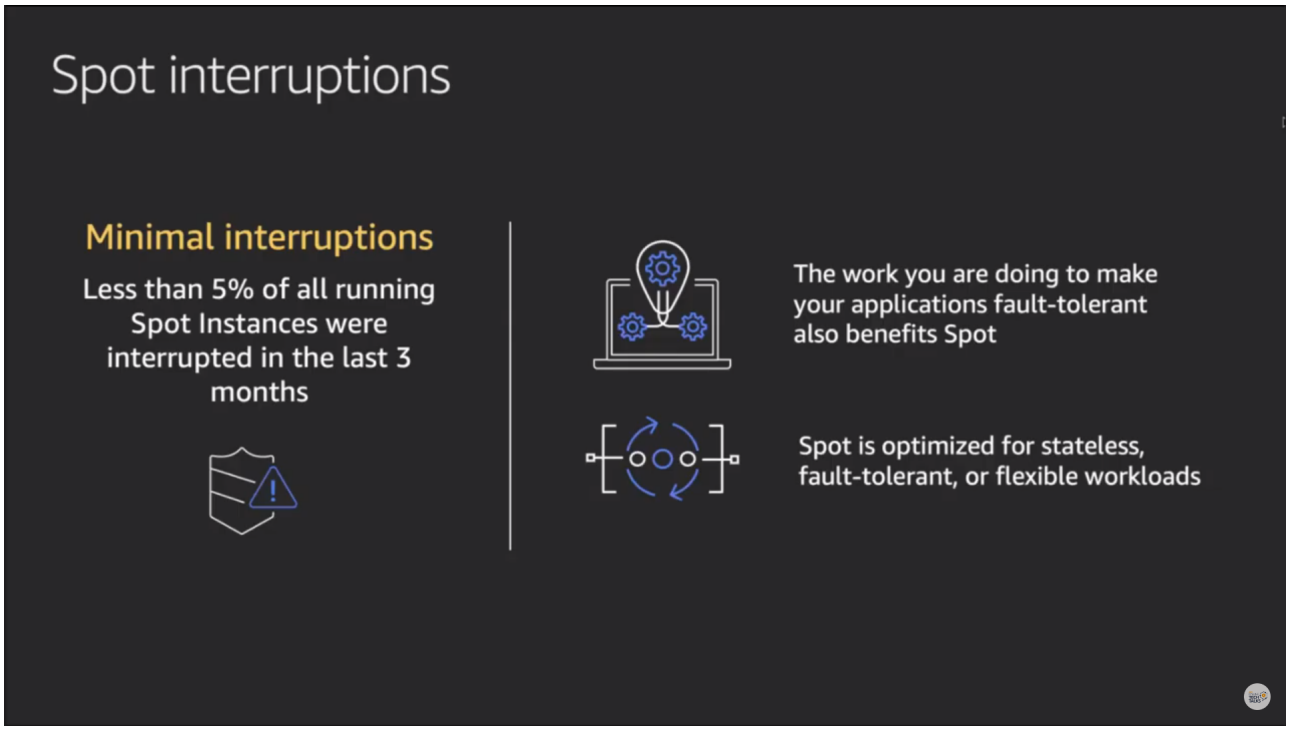


3. Why Spot Instances?

- uses spare EC2 capacity that is available for less than the On-Demand price
- batch allocation stategy includes spot capacity optimized, which allows AWS to find available capacity pools, and launch instances accordingly
- suitable for a hybrid compute environment such that if on-demand resource's threshold is met, scaling would shift towards using spot instances, which in turn will scale as per requirements without much worry on the cost.



- **spot interruptions:**
    - though spot interuptions may seem riskier at first, statistics show less than 5% of spot instance were interrupted in a span of 3 months



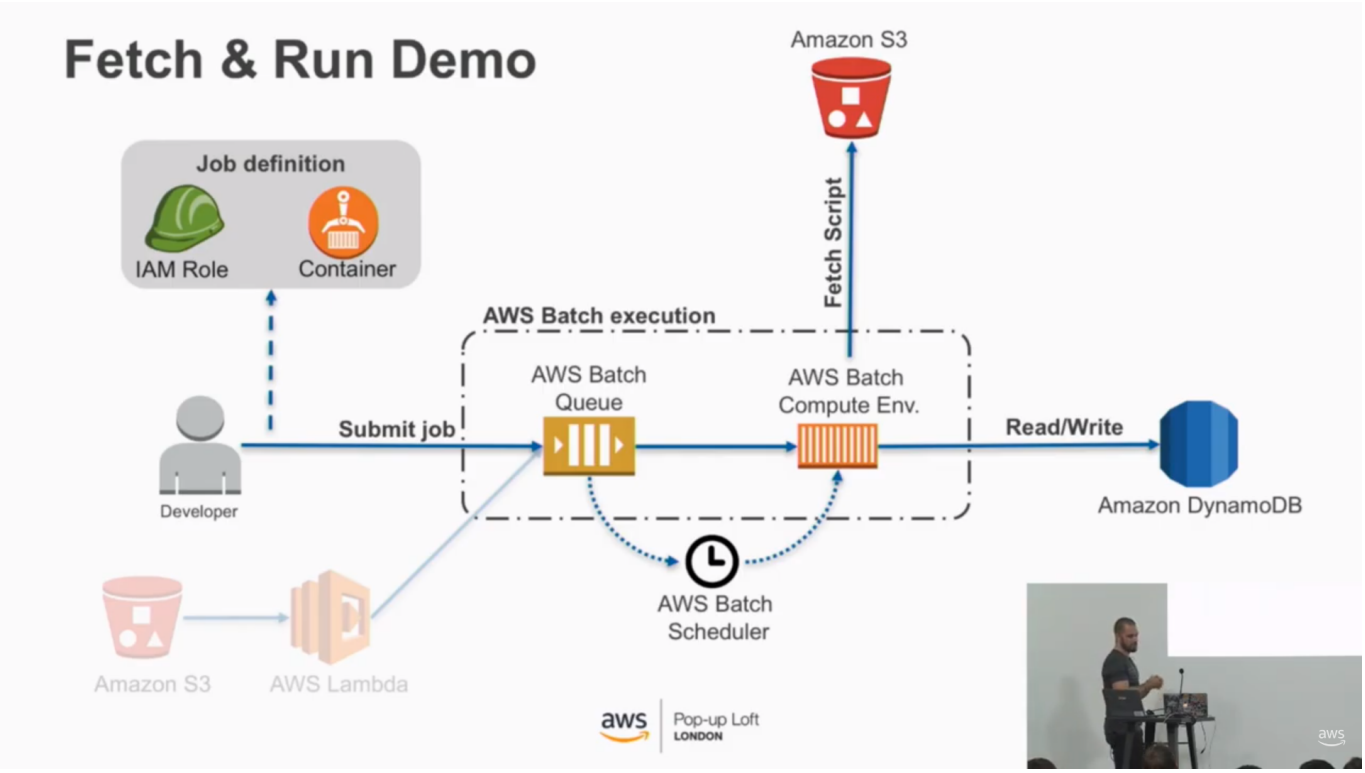
    - ECS agent in spot instances trigger a 2 minute notice prior to interruptions if the resources are required, however this is handled by:
        - monitoring spot instances for notices, which should drain instance containers and stop scheduling any jobs to that instance
        - allowing flexibility in terms of subnets and instance types leads to an increased combination of spot instance across resources

Best Practice Methodology

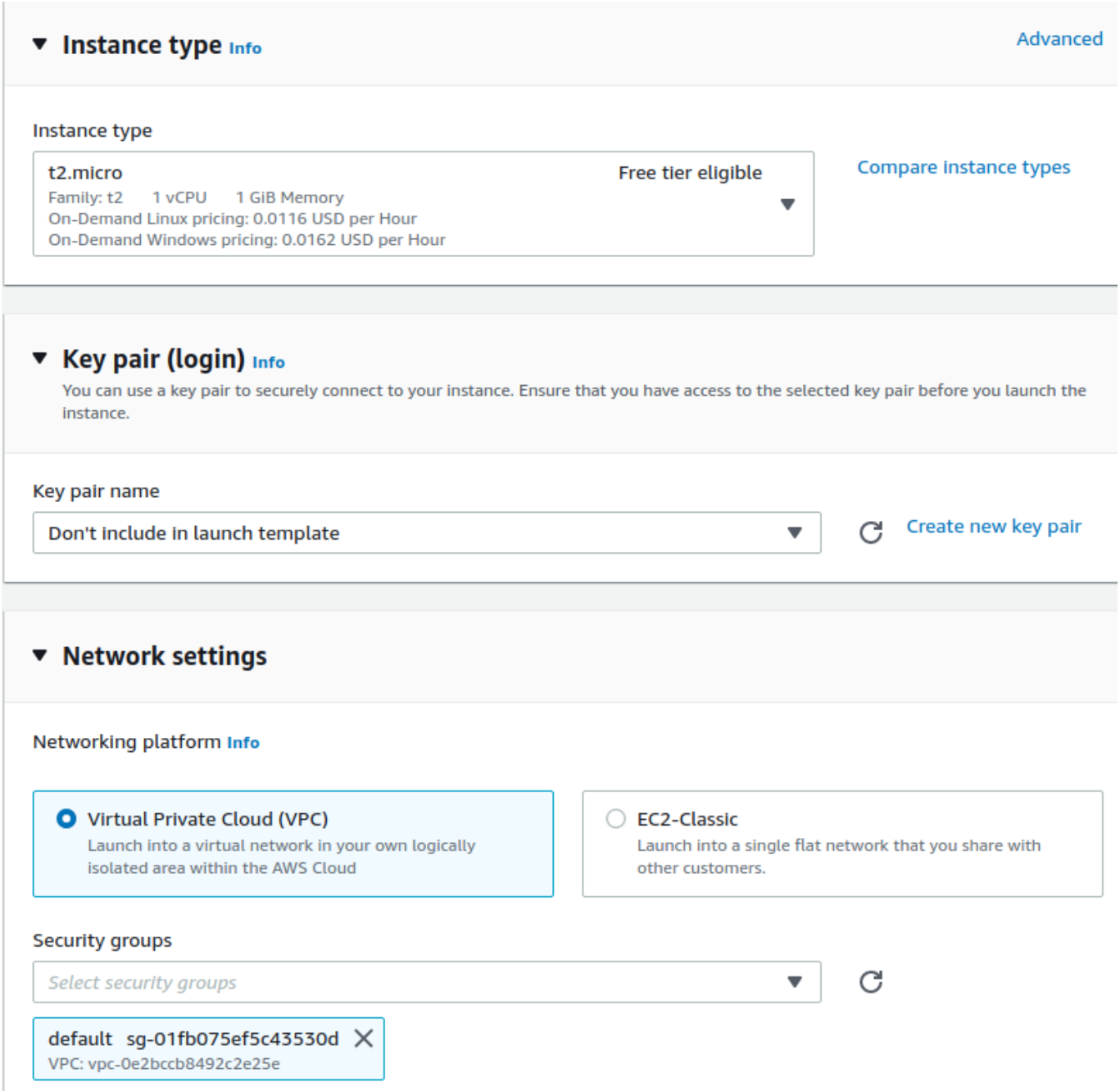
- Key steps:

    - ☐ Build a Docker image with the fetch & run script
    - ☐ Create an Amazon ECR repository for the image
    - ☐ Push the built image to ECR
    - ☐ Create a simple job script and upload it to S3
    - ☐ Create an IAM role to be used by jobs to access S3
    - ☐ Create a job definition that uses the built image
    - ☐ Submit and run a job that execute the job script from S3

The figure below shows the idea architecture for the solution, which is mainly to have a script that fetches jobs to run from S3, batch processess the data, and have the ready data written to a DB.



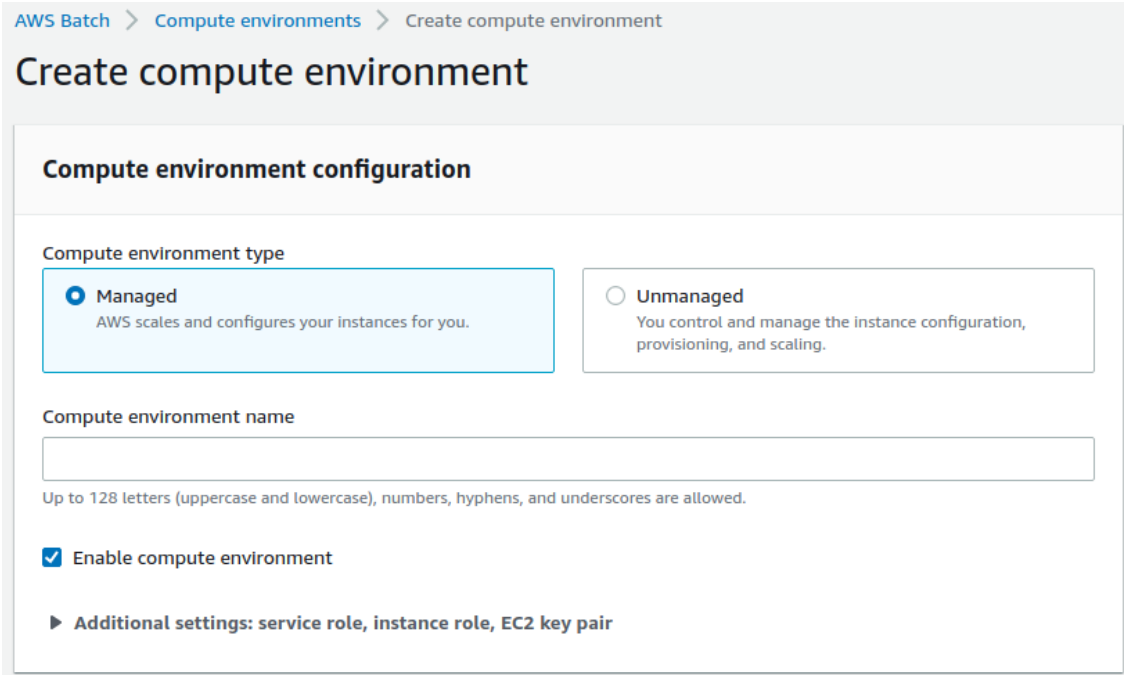
- create EC2 template and enable spot interruption handling

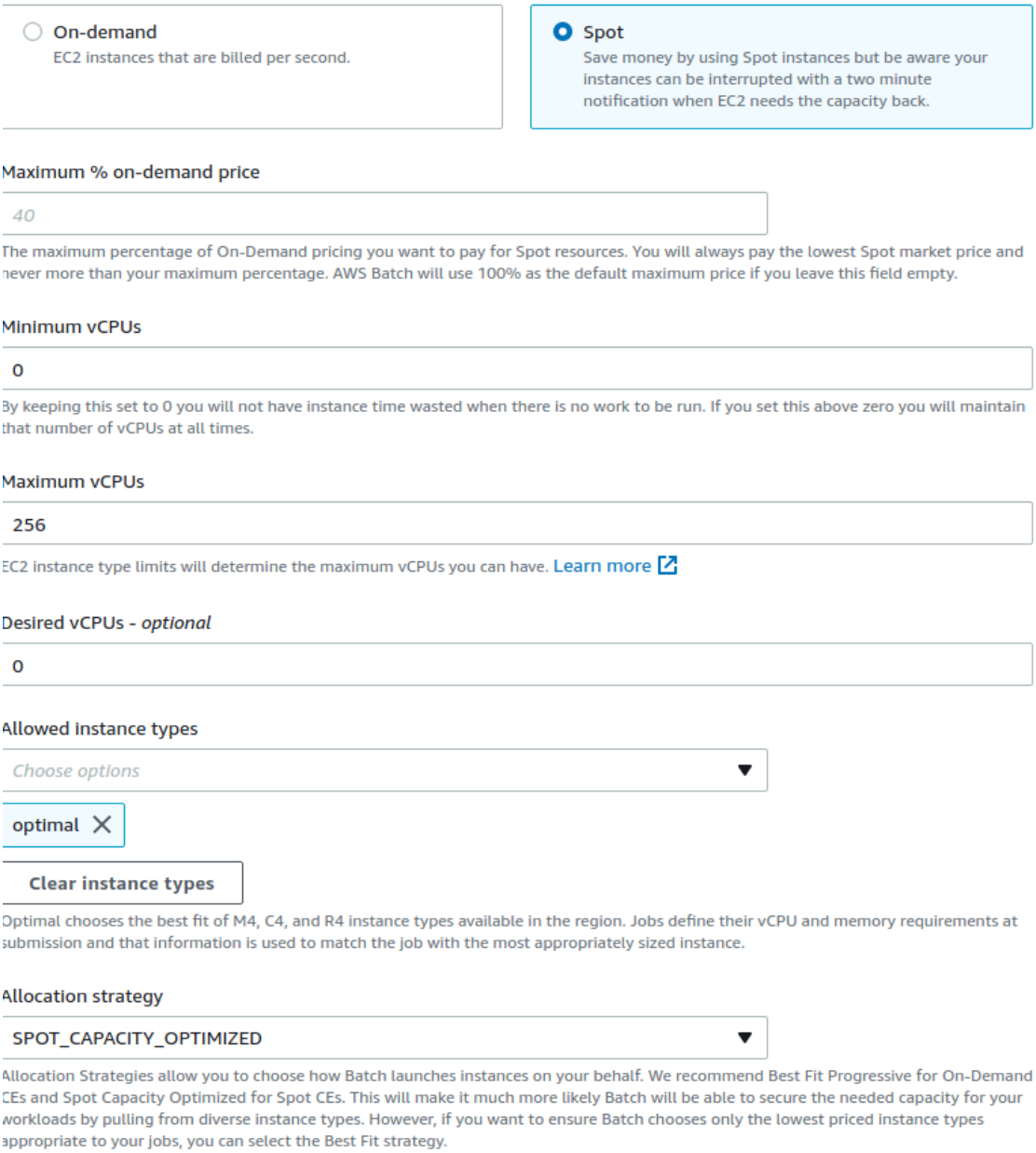


- job retries enabled
- **Handling Spot Interruption:**

  Advanced Settings > User Data > `echo "ECS_ENABLE_SPOT_INSTANCE _DRAINING=true" >> /etc/ecs/ecs.config`

- AWS Batch > create
    - compute environment





- job queue
- job definition

- jobs

Python

Source code and documentation can be found here