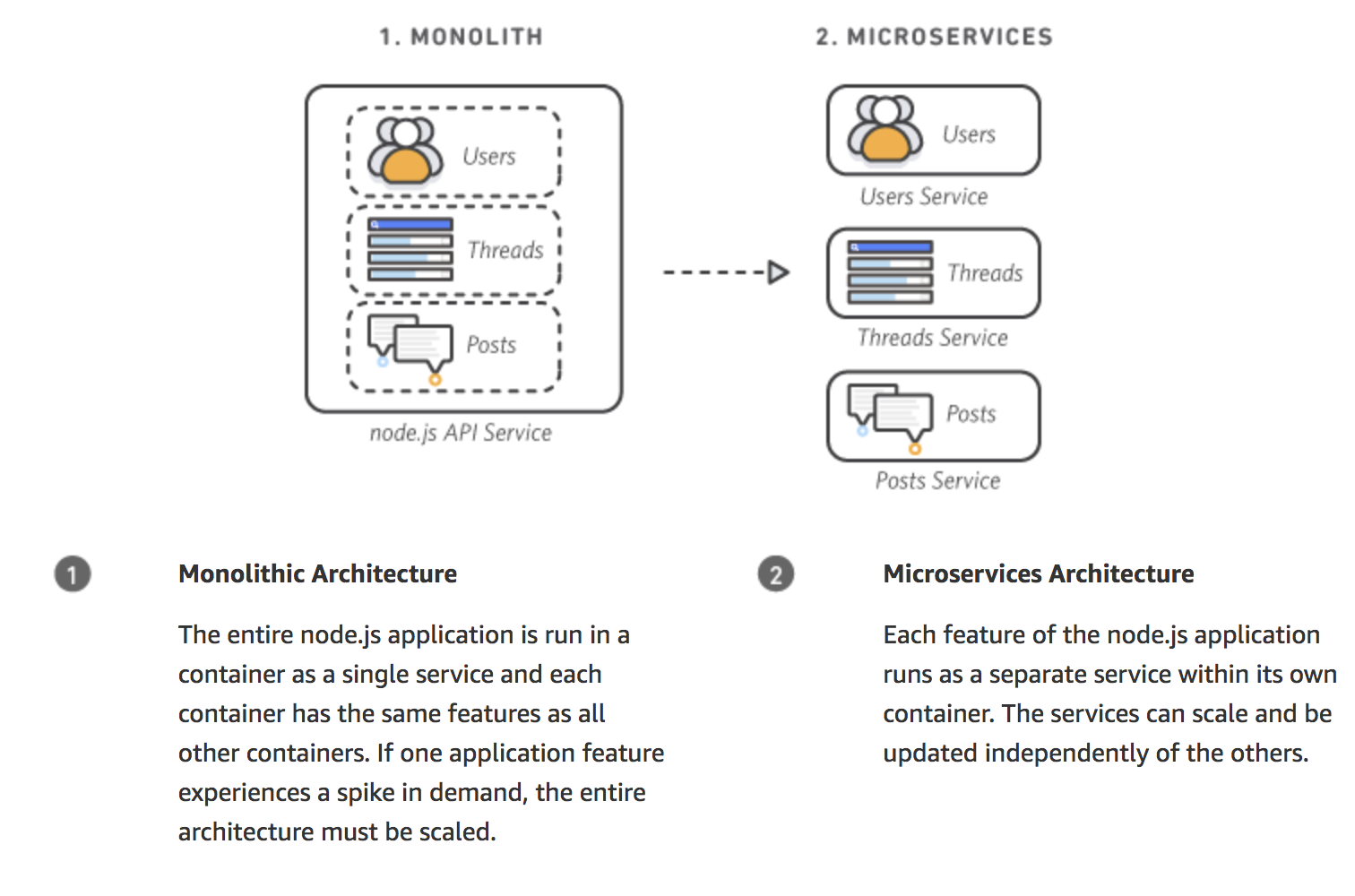
# **AWS Project :** **Break a Monolith Application into Microservices**

Time spent: 200min

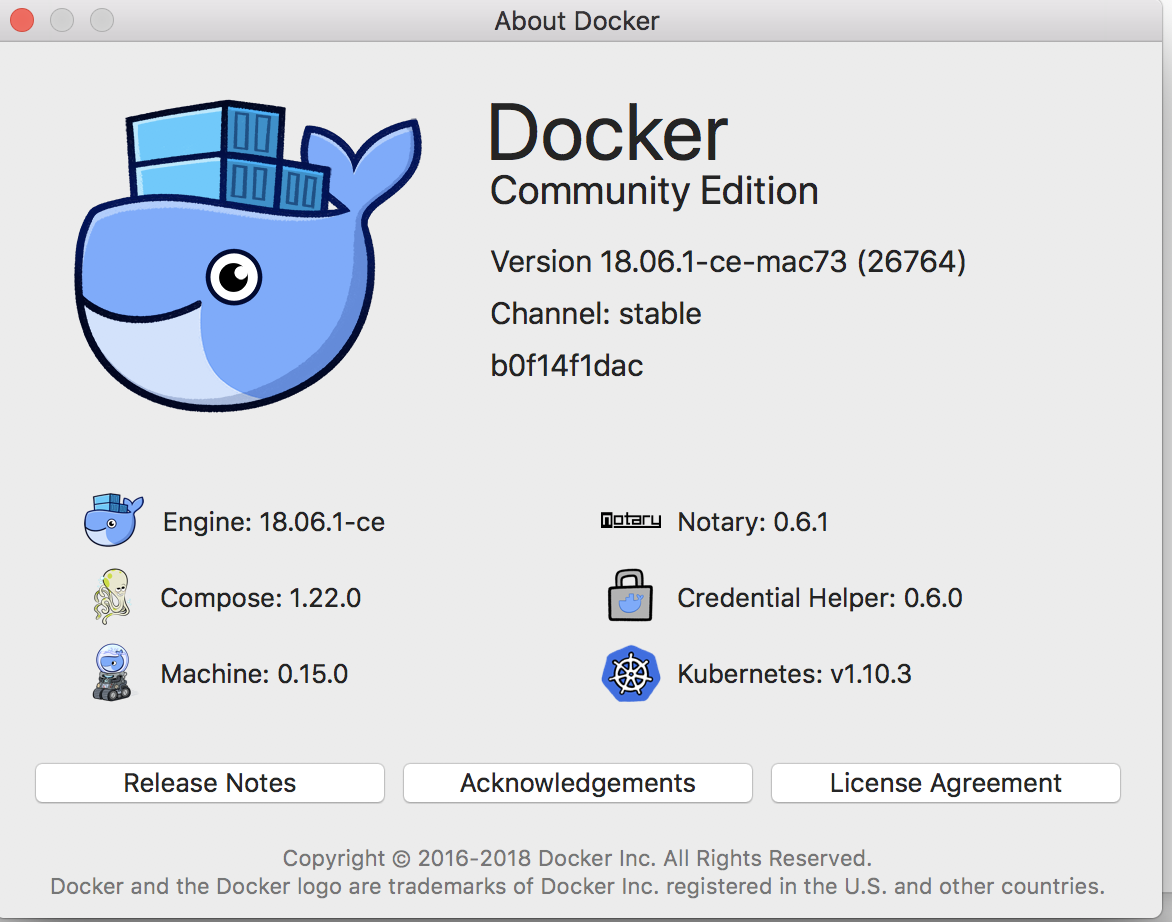
1. Overview

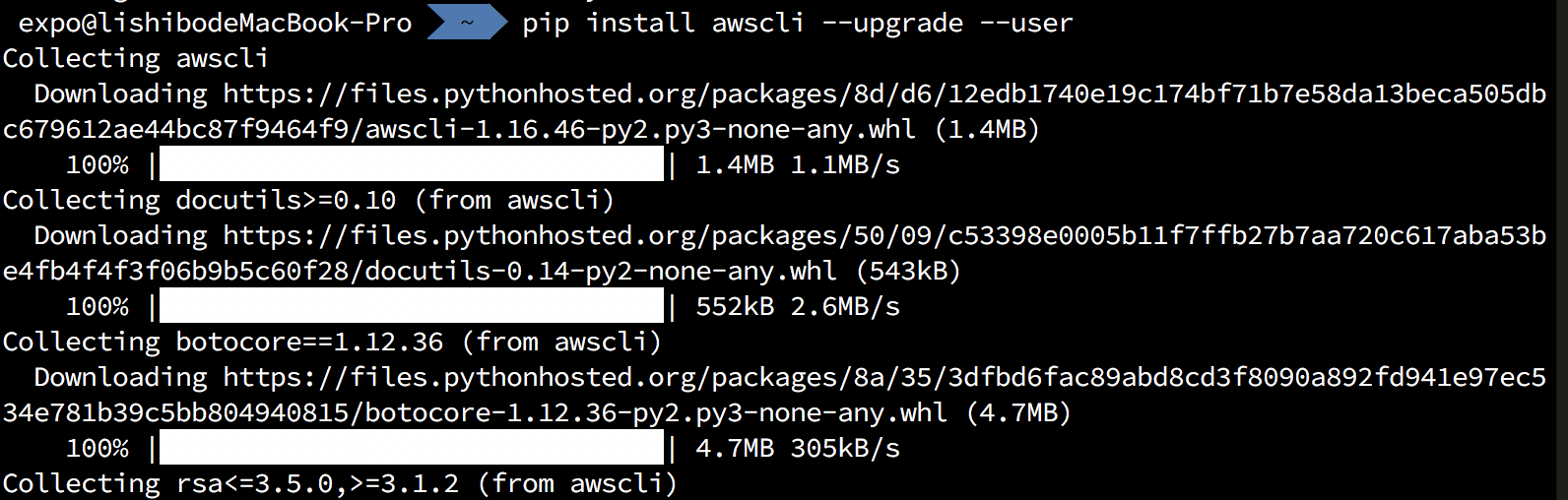
In this tutorial, you will deploy a monolithic node.js application to a Docker container, then decouple the application into microservices without any downtime. The node.js application hosts a simple message board with threads and messages between users.



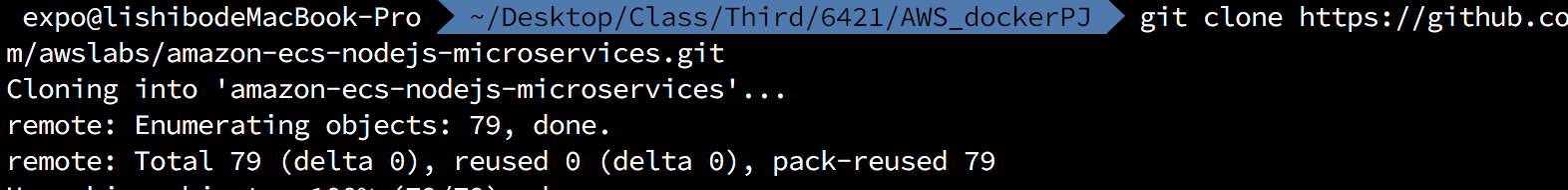
1. Containerize the Monolith
2. Environment set up

AWS CLI, Docker

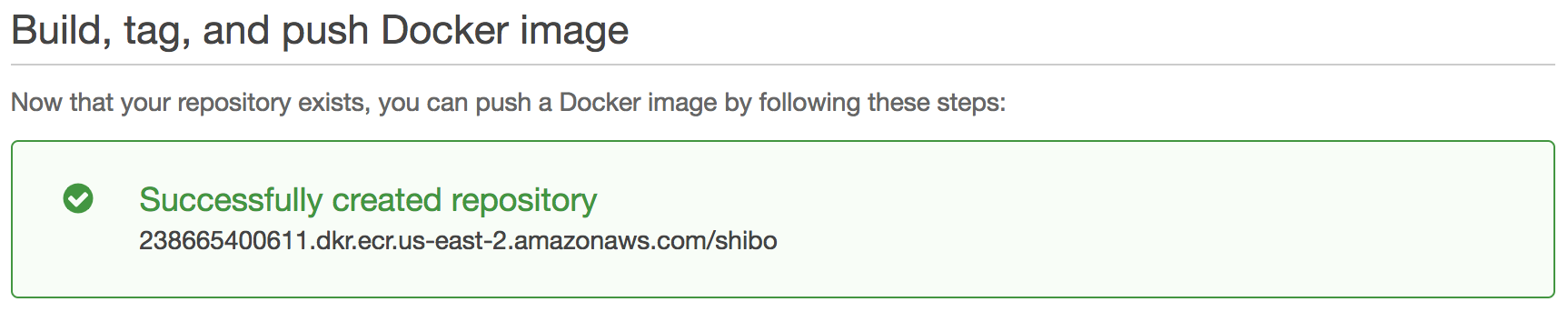




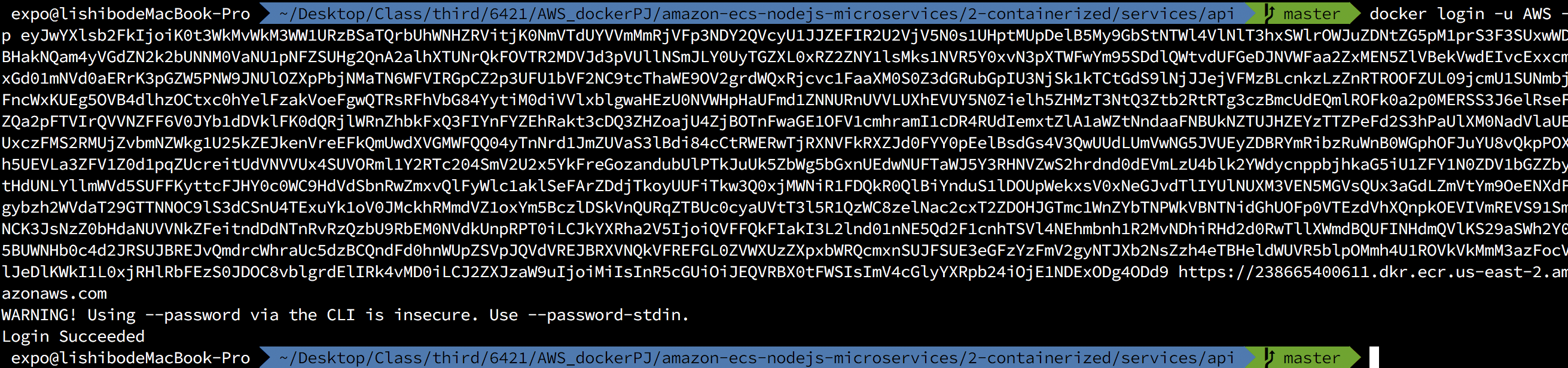
1. Get project



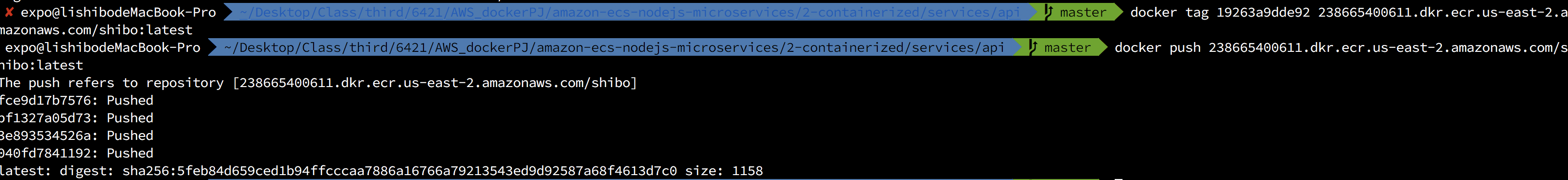
1. Get repository

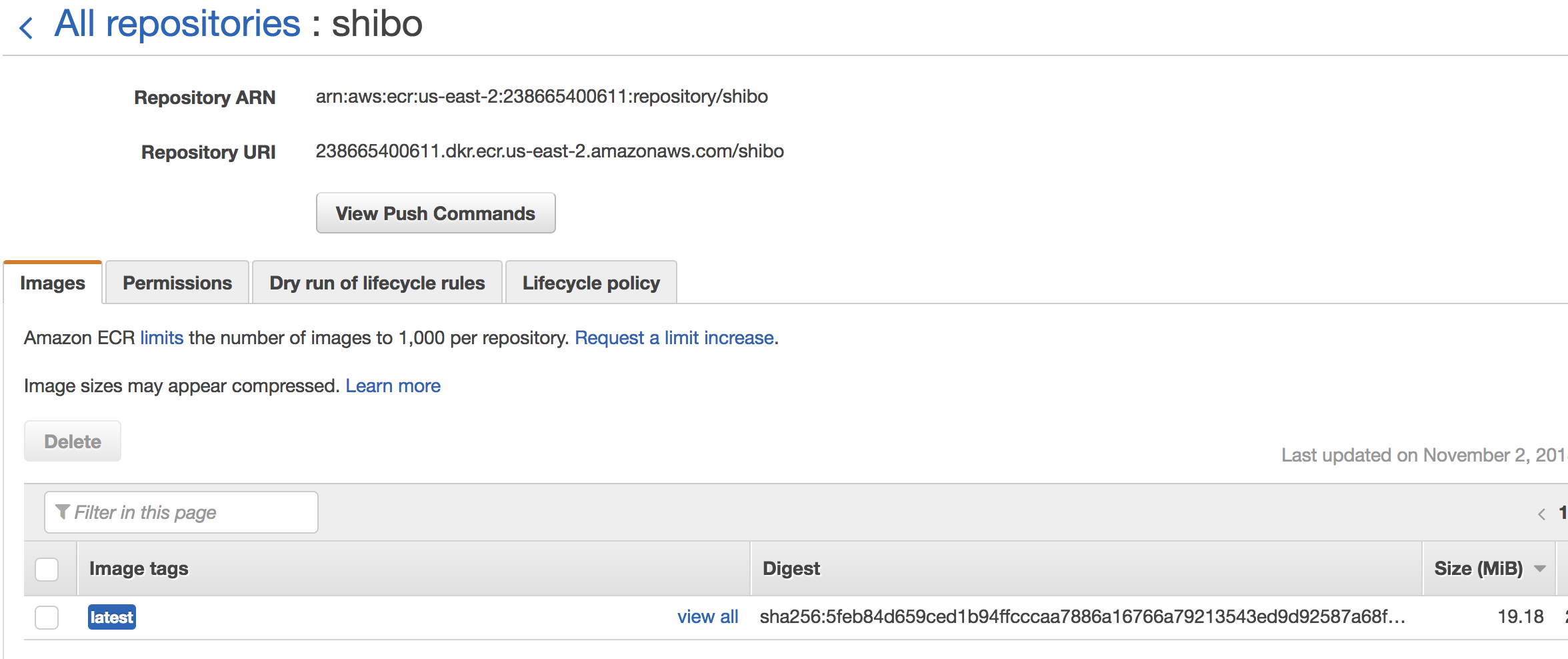


1. Get docker login

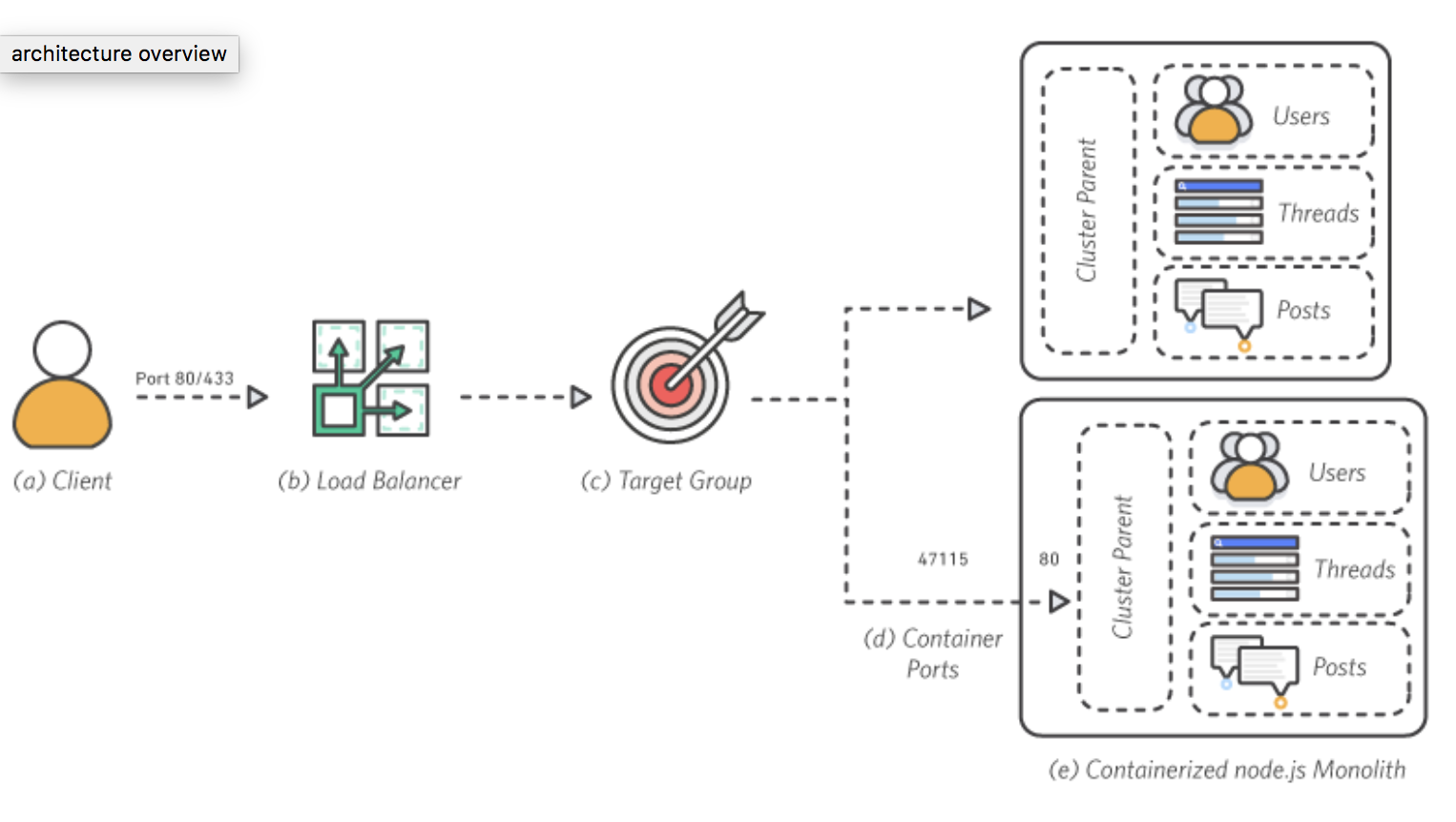


1. Push the Image

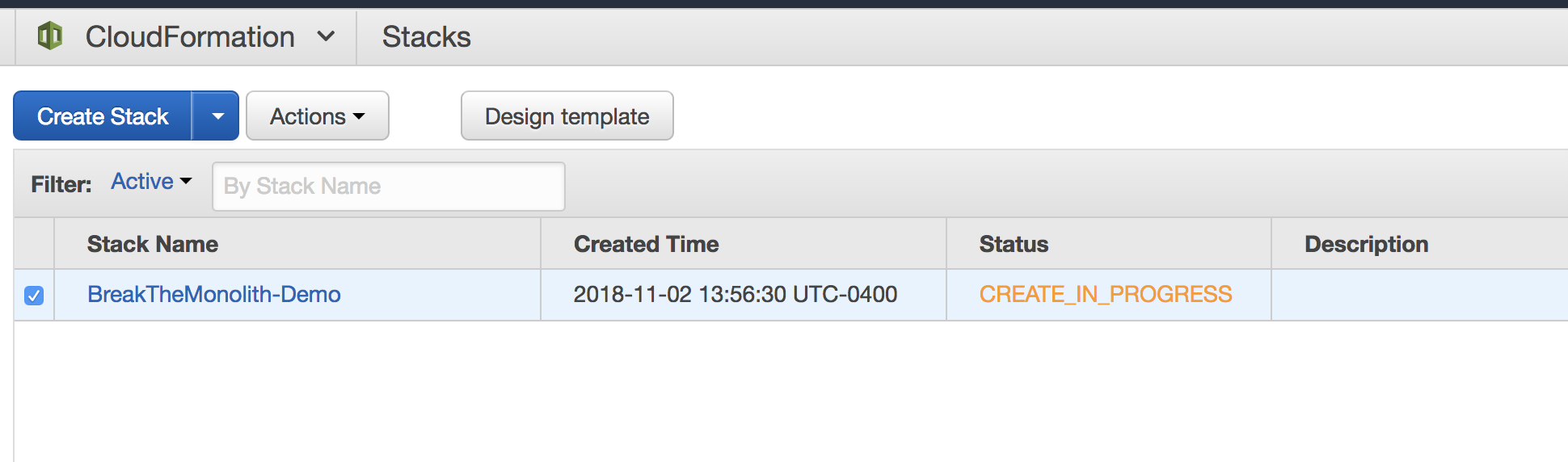




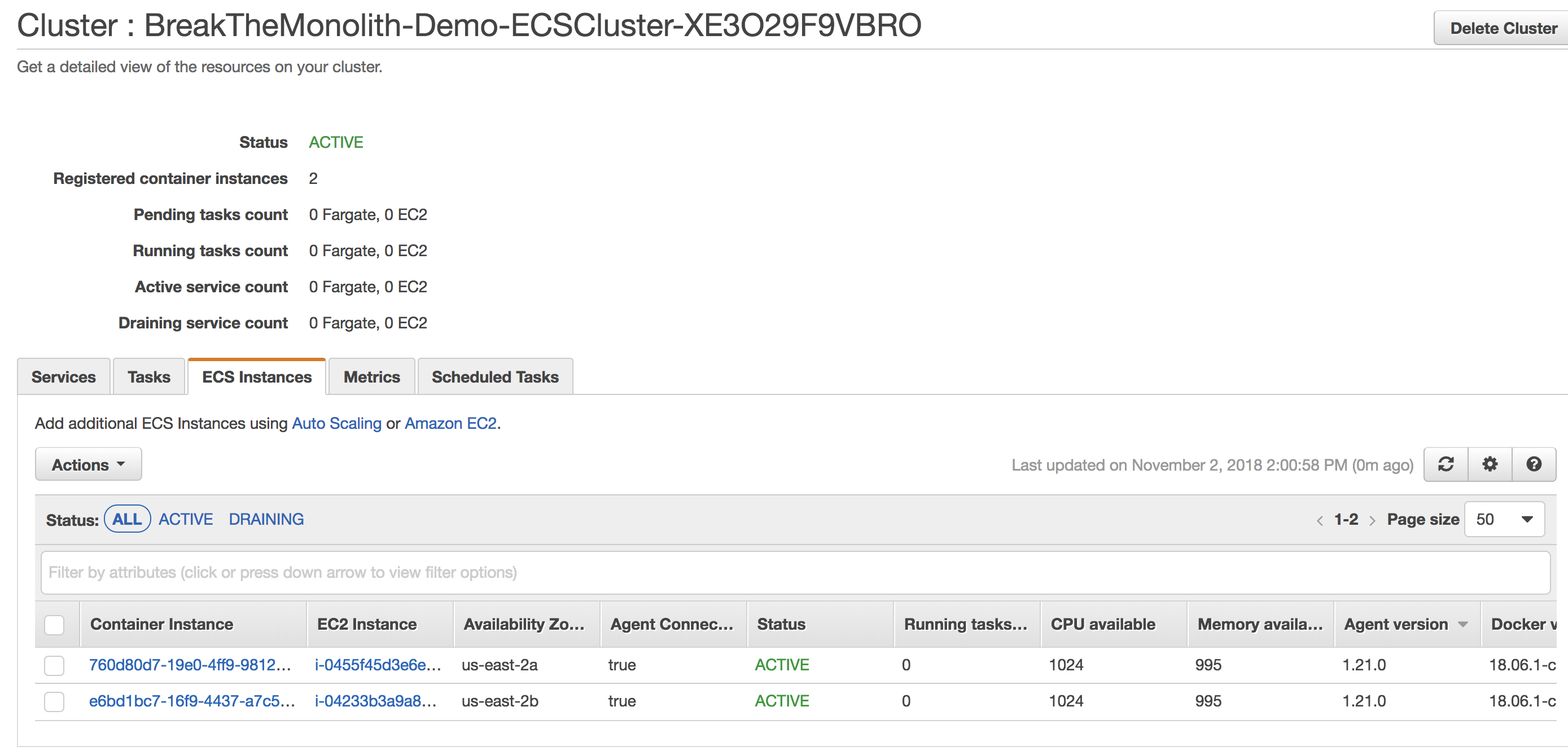
1. Deploy the Monolith
2. Overview



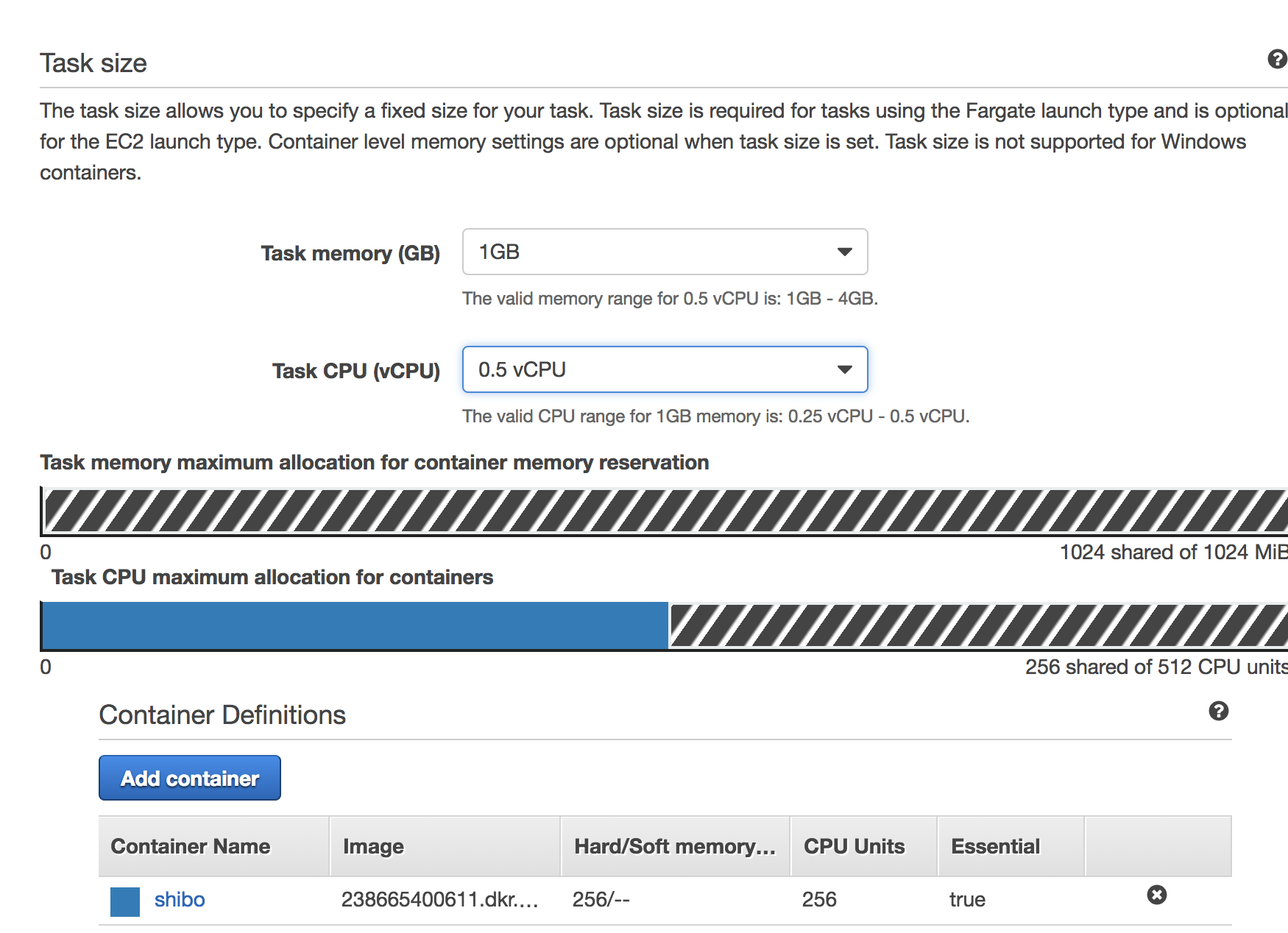
1. Create the Stack



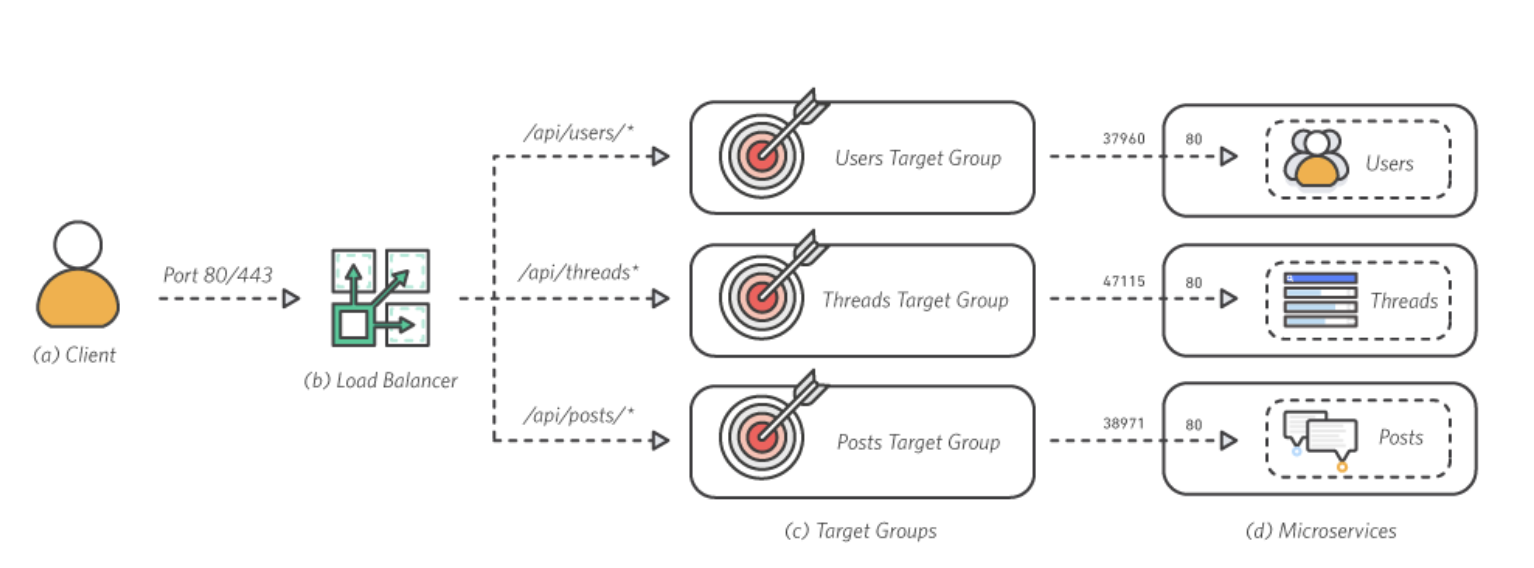
1. Check cluster is running



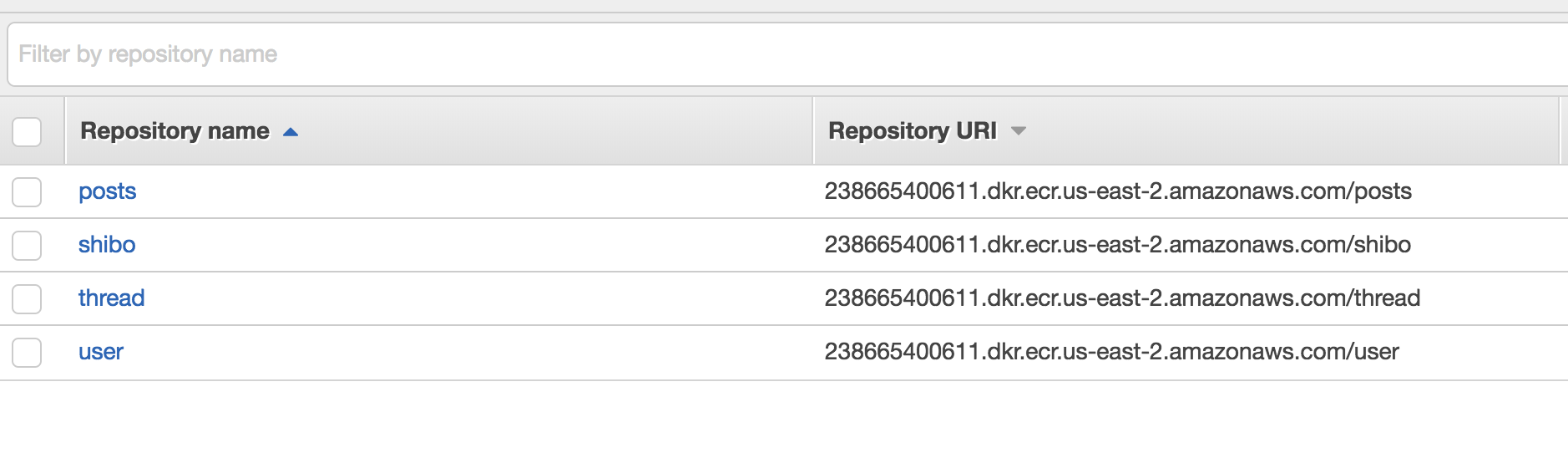
1. Define the task



1. Break the Monolith
2. Overview



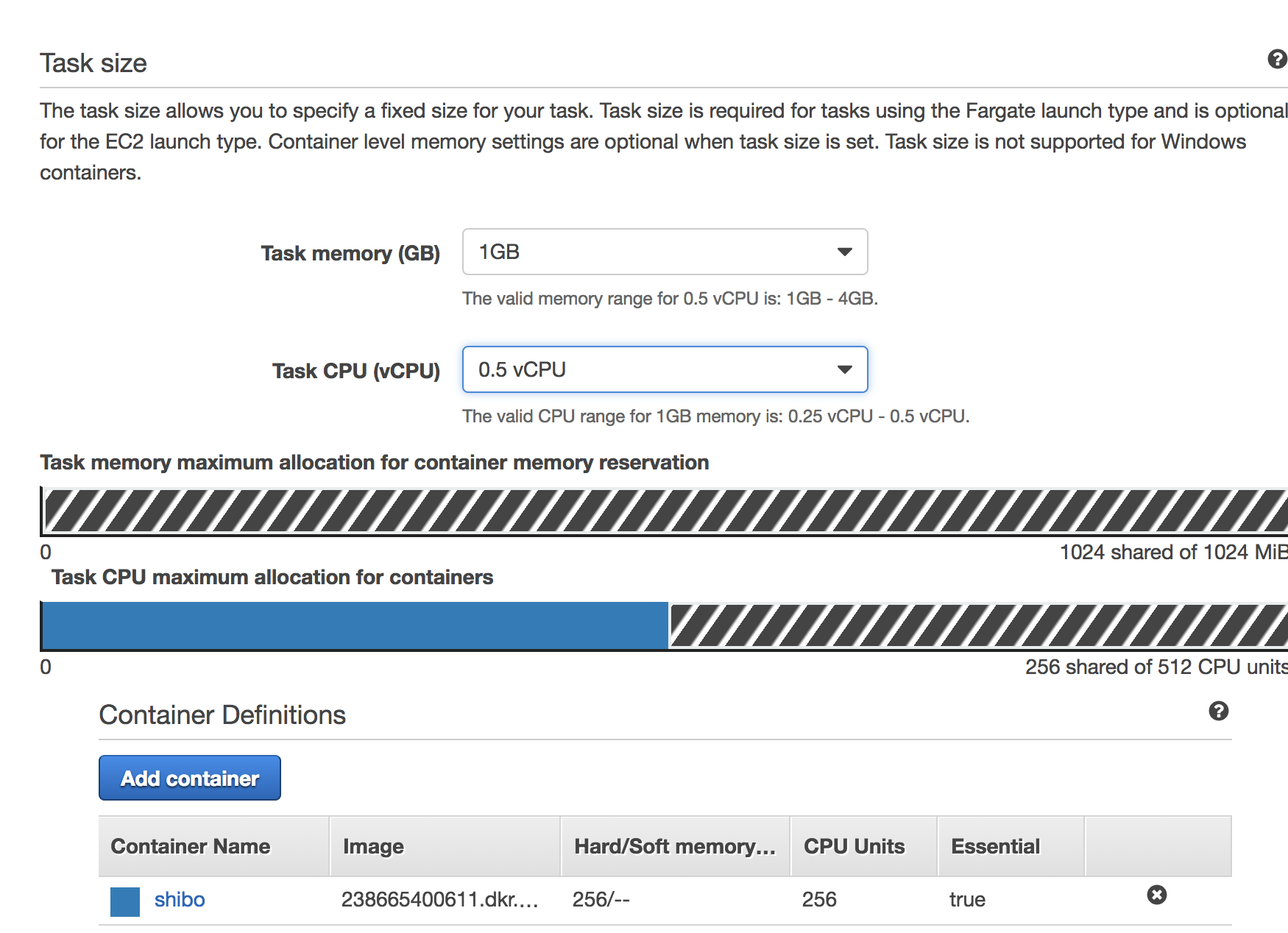
1. Create 4 repositories



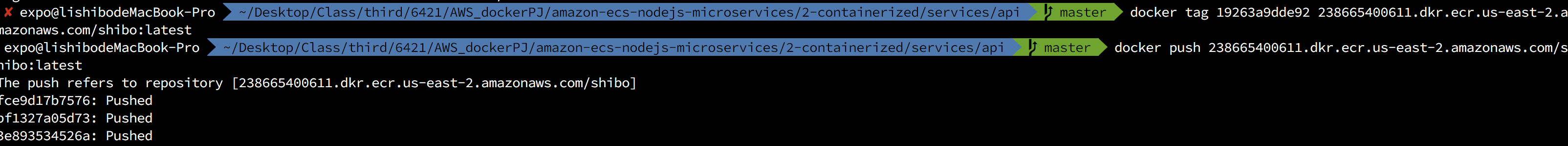
1. Tag and push each image
2. Deploy Microservices
3. Architecture overview

In this module, you will deploy your node.js application as a set of interconnected services behind an Application Load Balancer (ALB). Then, you will use the ALB to seamlessly shift traffic from the monolith to the microservices.

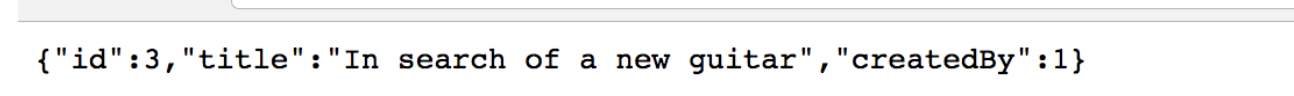
1. Define the task



1. Start Microservices



1. Validate the deployment



**Summary**

I have harvested a lot of precious experience about docker and containers in this project on AWS. Following the instruction, not only do I got the deeper understanding of dockers and containers, but also I connected the knowledge with the real usage.