Break a Monolith Application into Microservices

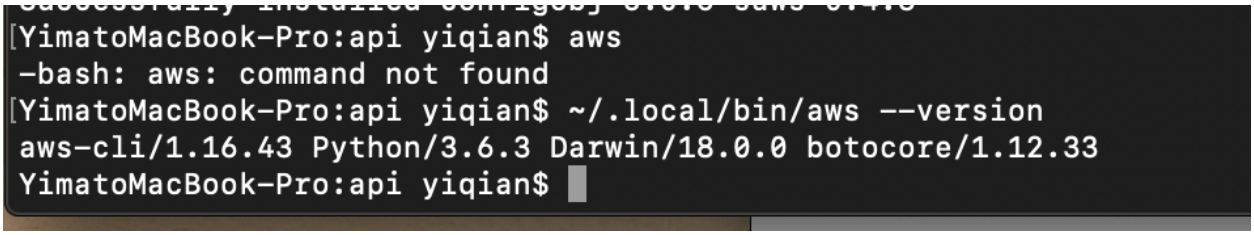
Overview

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.

Module 1 Containerize the Monolith

Step 1 Get Setup

1. Have AWS account
2. Install Docker
3. Install the AWS CLI
   after using pip install awscli —upgrade —user, use aws —version to verify. Shows aws command not found. Try another:



4. Have a Text Editor

Step 2 Download & Open the Project

Download from GitHub

Step 3 Provision a Repository

The repository address

Step 4 Build & Push the Docker Image

1. Configure your credentials

```
YimatoMacBook-Pro:api yiqian$ ~/.local/bin/aws configure
AWS Access Key ID [None]: AKIAJZITGLEDT7MLXEBQ
AWS Secret Access Key [None]: DEciIllb+bA38MEB2W92WeChDc0WdiLbJVHqlQIm
Default region name [None]: us-east-2
Default output format [None]: json
YimatoMacBook-Pro:api yiqian$
```

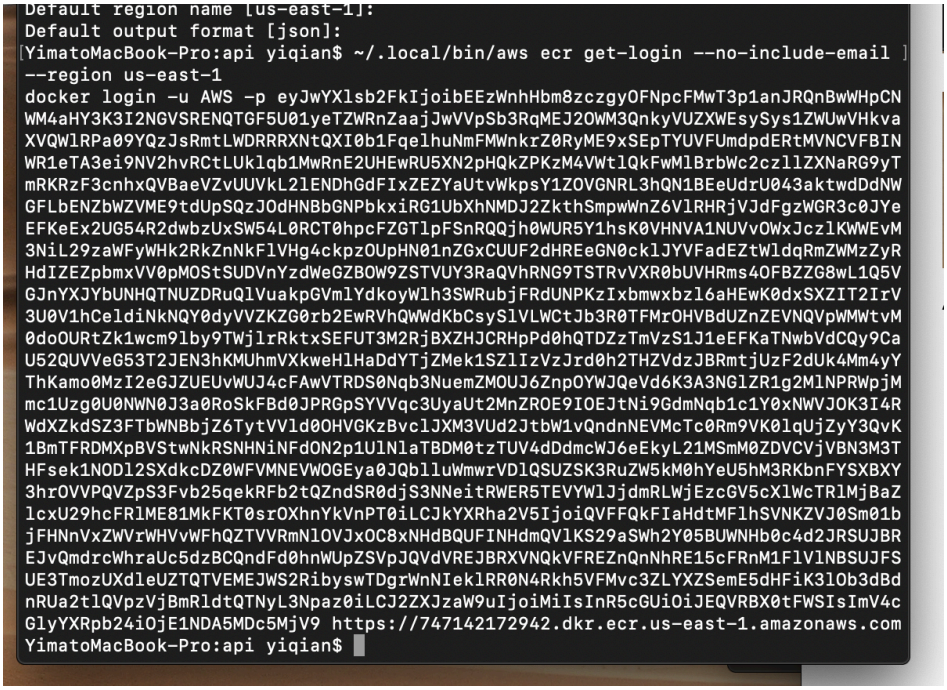
```
[YimatoMacBook-Pro:api yiqian$ ~/.local/bin/aws ecr get-login --no-include-email
--region us-west-2

An error occurred (UnrecognizedClientException) when calling the GetAuthorizatio
nToken operation: The security token included in the request is invalid.
YimatoMacBook-Pro:api yiqian$ ~/.local/bin/aws ecr get-login --no-include-email
--region us-west-2
```

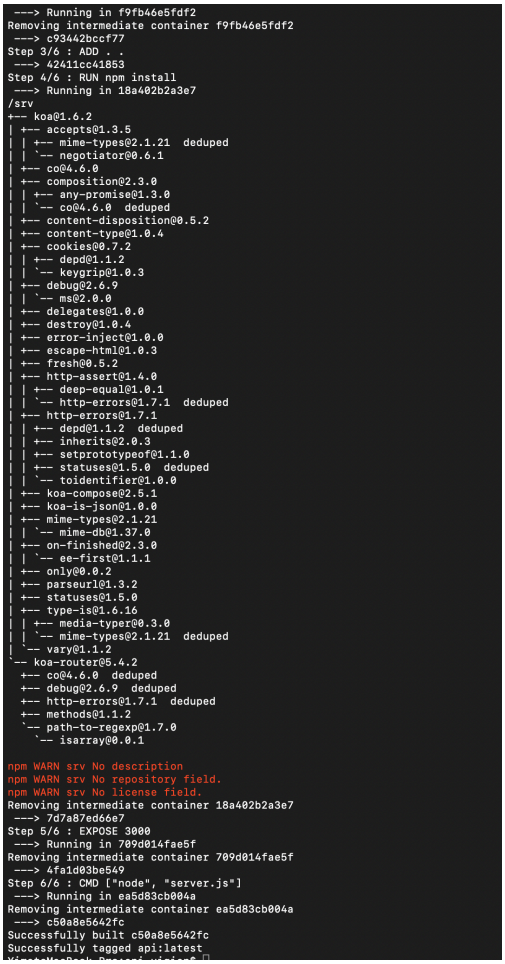
Add json permission in User

策略摘要 | {} JSON | 编辑策略

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage"
            ],
            "Resource": "*"
        }
    ]
}
```



Build the Image



Push the image to ECR:

```
The push refers to repository [747142172942.dkr.ecr.us-east-1.amazonaws.com/api]
002b83738171: Layer already exists
d1c9a2ba6d79: Layer already exists
3e893534526a: Layer already exists
040fd7841192: Layer already exists
latest: digest: sha256:6e3fdd2228a35326b24f148f697739d72b61675743bc85de79c28526918795b5 size: 1158
YimatoMacBook-Pro:api yiqian$
```

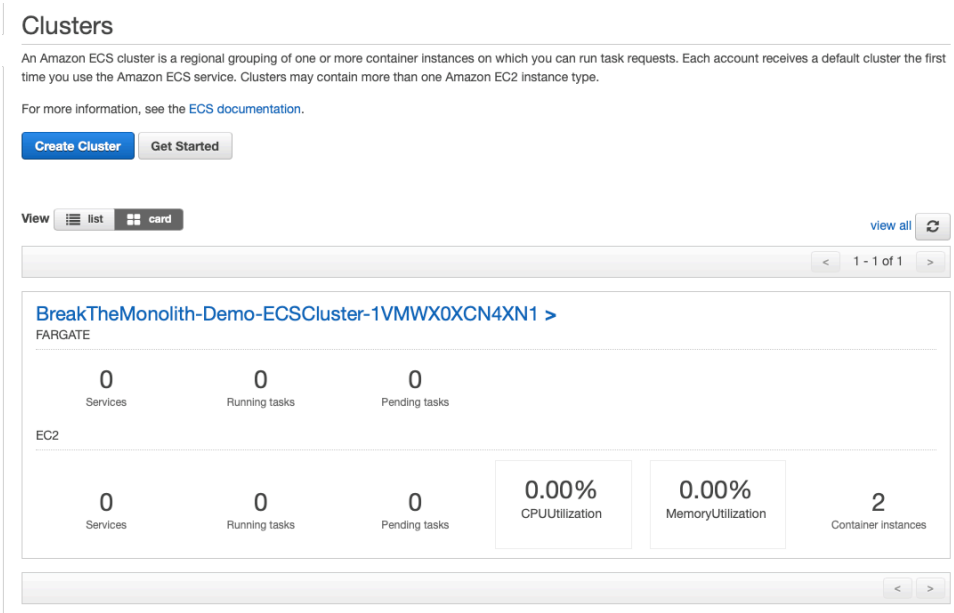
Module 2 Deploy the Monolith

Step 1 Launch an ECS Cluster using AWS CloudFormation

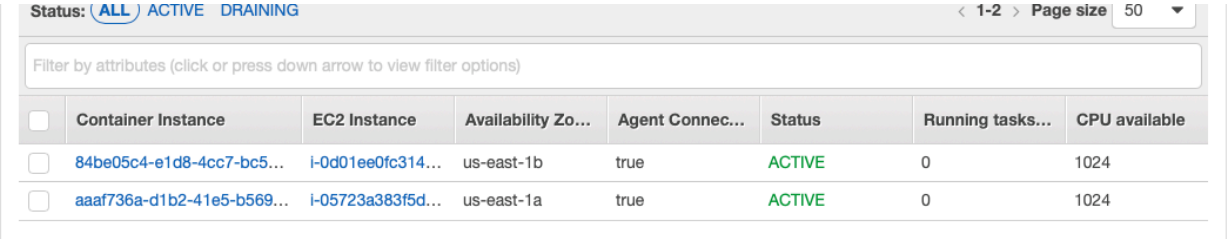
Create stack

Step 2 Check your Cluster is Running

Cluster

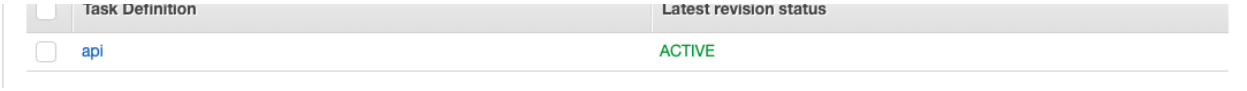


2 EC2 instances



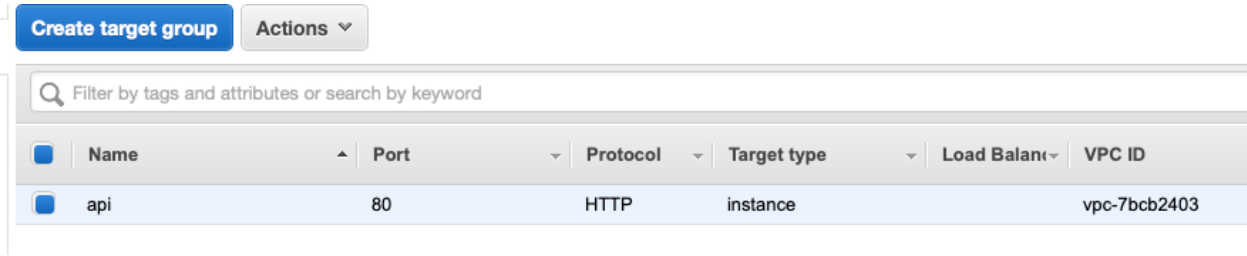
Step 3 Write a Task Definition

Create task definition



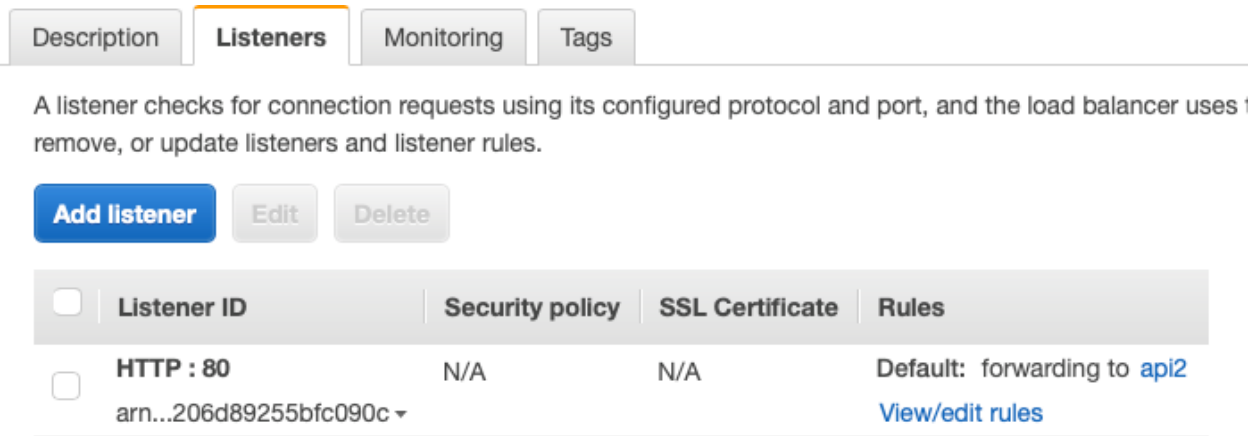
Step 4 Configure the Application Load Balancer: Target Group

Configure the ALB Target Group



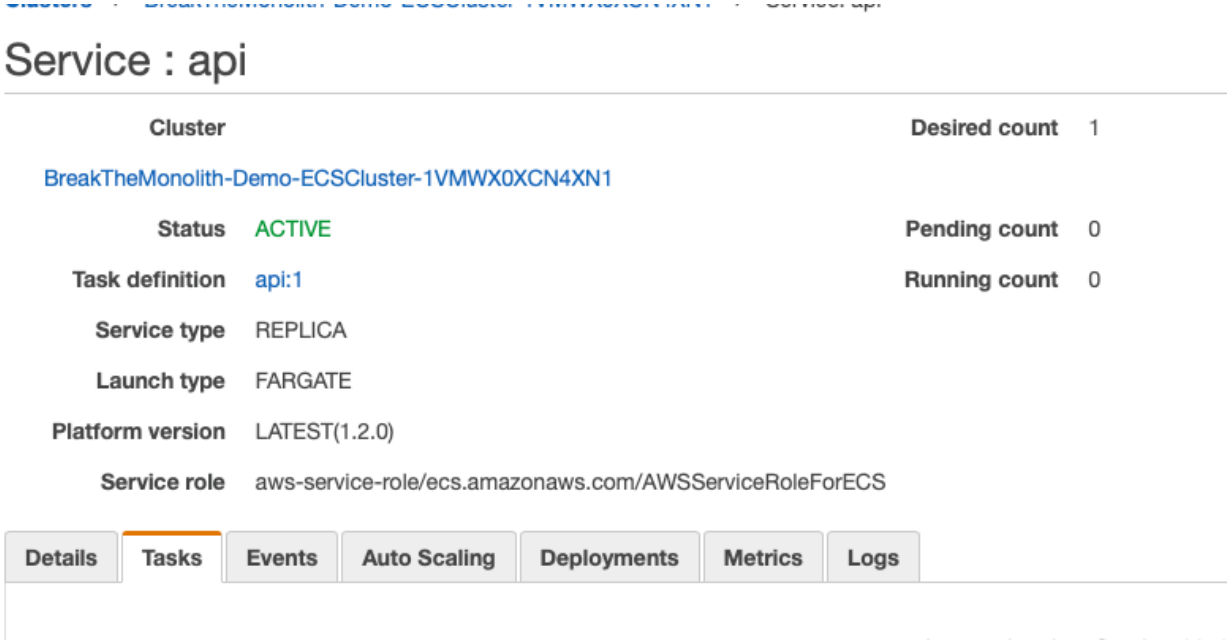
Step 5 Configure the Application Load Balancer: Listener

Add listener

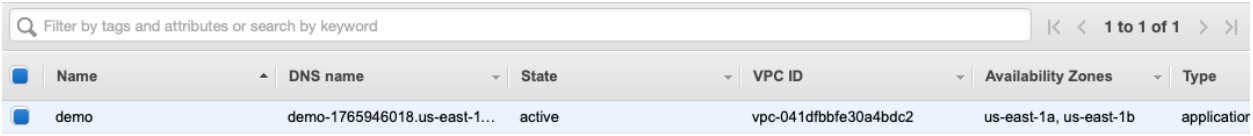


Step 6 Deploy the Monolith as a Service

Create Service



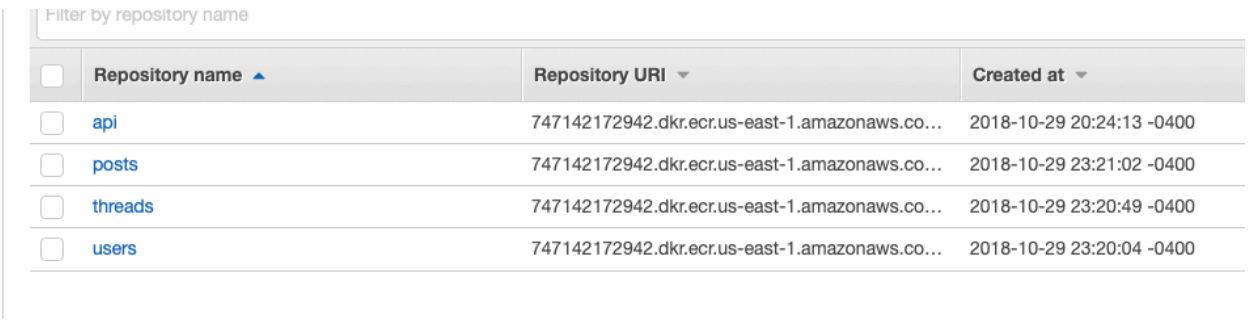
Step 7 Test your Monolith



Part 3 Break the Monolith

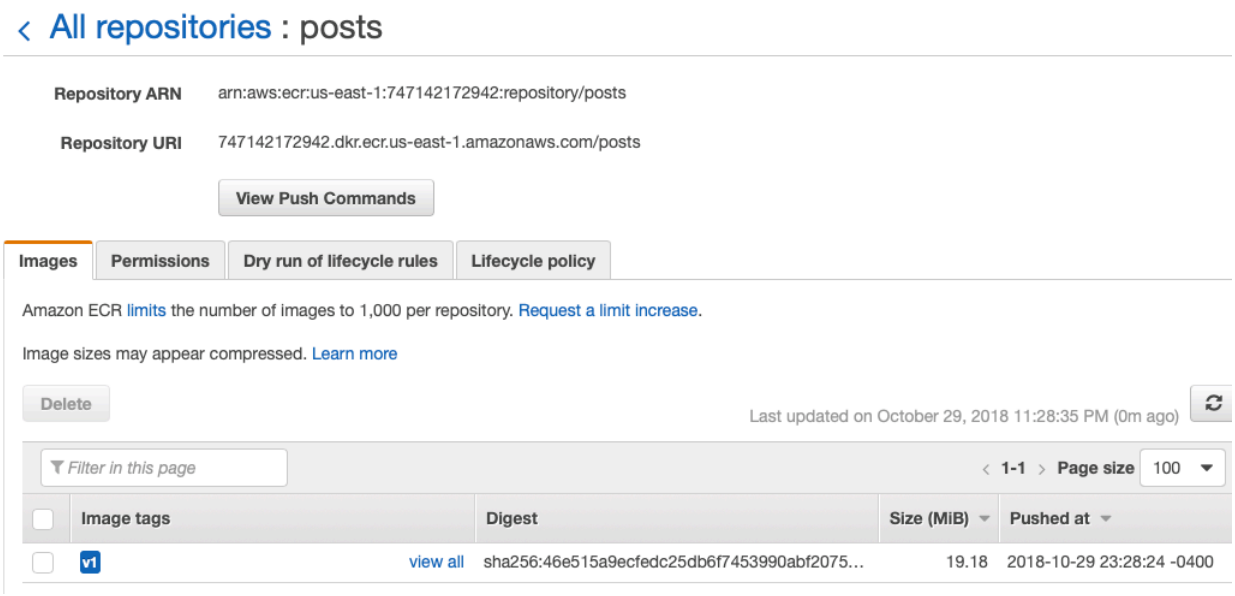
Step 1 Provision the ECR Repositories

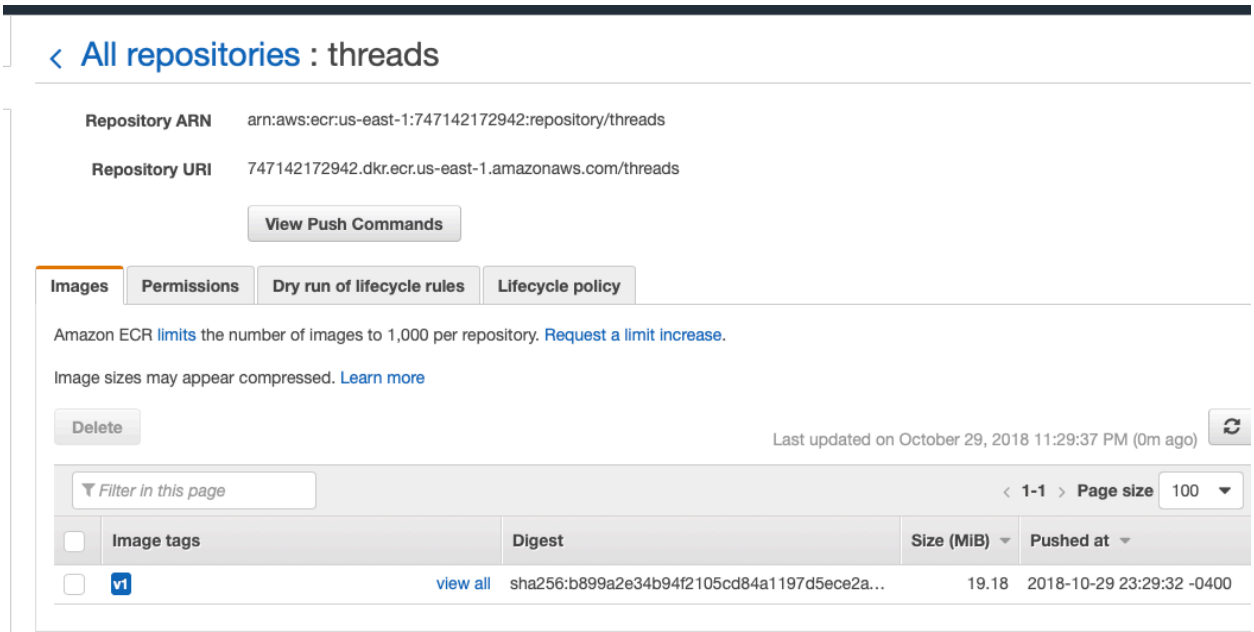
Create repository

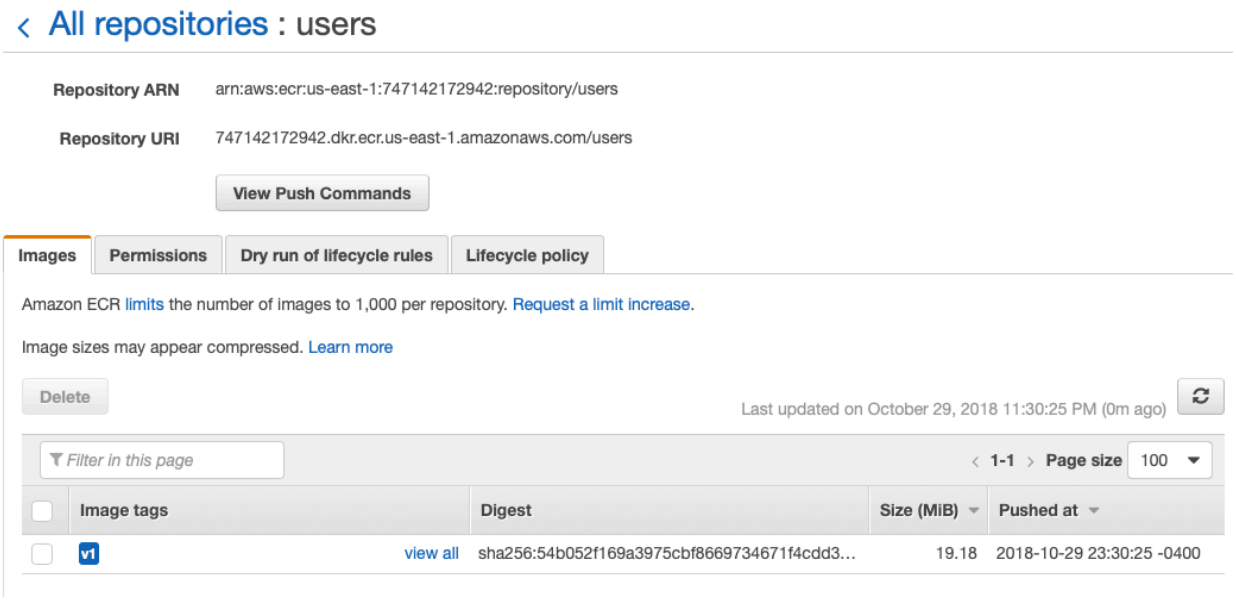


Step 3 Build and Push Images for Each Service

Build images tag then push

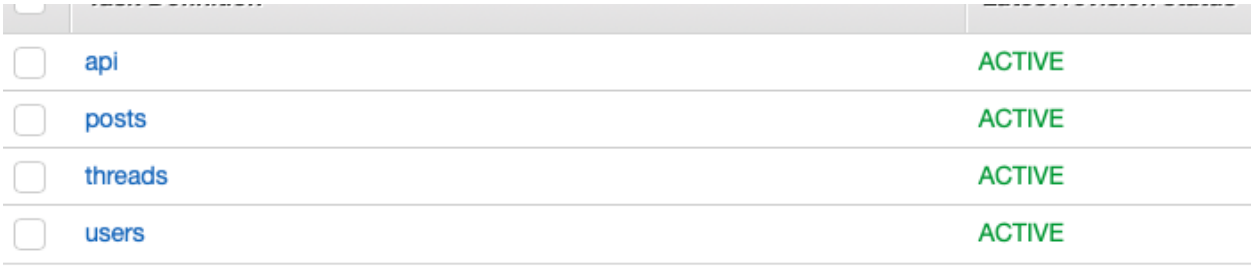




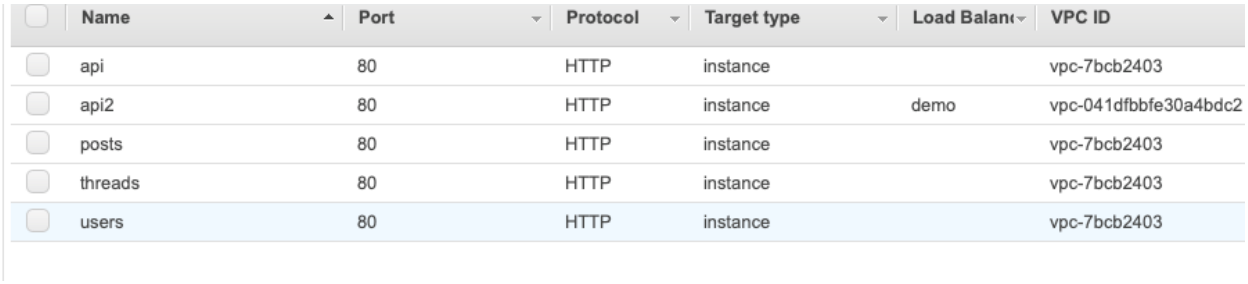


Part 4 Deploy Microservices

Step 1 Write Task Definition for your Services

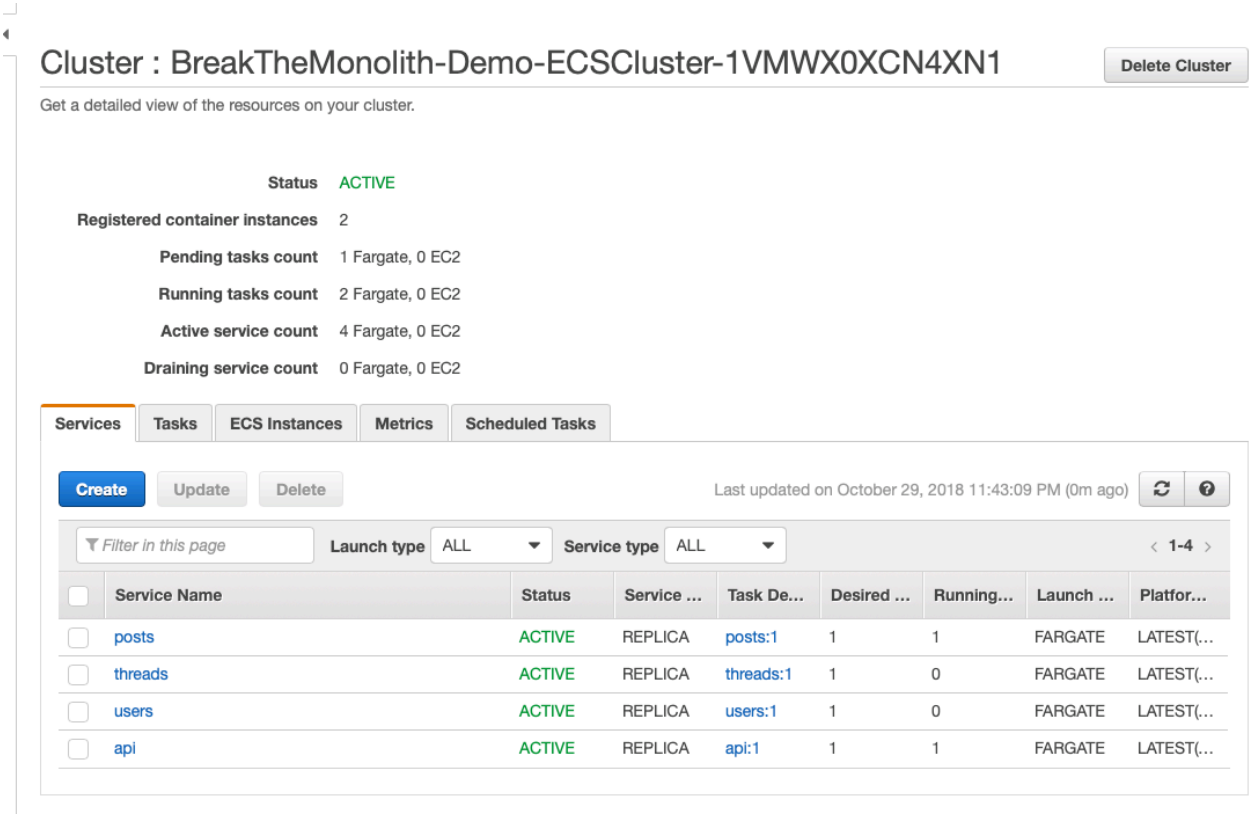


Step 2 Configure the Application Load Balancer: Target Groups



Step 3 Configure Listener Rules

Step 4 Create Services



Step 6 Validate your Deployment


```
{"id":3,"title":"In search of a new guitar","createdBy":1}
```