**Setup on AWS**

You can deploy Docker containers using the Amazon EC2 Container Service –

<https://us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/getStarted>

**1. Sign up for the AWS Free Tier**

A/C ID - 664375028740

**2. Get setup in ECR as per**

<http://docs.aws.amazon.com/AmazonECR/latest/userguide/get-set-up-for-amazon-ecr.html>

(NOTE – Have I used this account – NOT YET).

IAM User Details

jimmyc

F}yPh=Ub[u+2

**3. Install and configure the AWS CLI**

<http://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html#install-msi-on-window>

Create a new access key via the Security Credentials page in AWS:

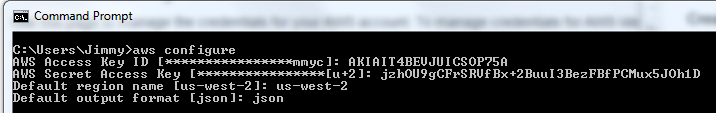
Access Key ID:

AKIAIT4BEVJUICSOP75A

Secret Access Key:

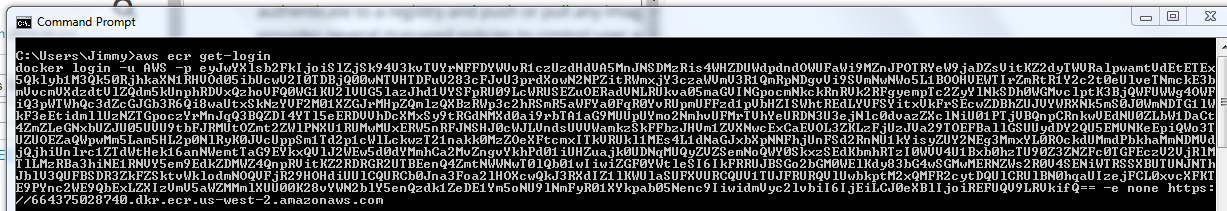
jzhOU9gCFrSRVfBx+2BuuI3BezFBfPCMux5JOh1D

Run ‘aws configure’ from the command line:



This gives us the command we need later to authenticate our docker client with the Amazon ECR registry:

(Authentication tokens are valid for 12 hours)



**Actual Deployment**

Ref: <http://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>

(NOTE - Must fix issues with IP's in the services before I deploy them)

1. Create a new repository that will store the containers - called it 'cloudev'

Repository URL - 664375028740.dkr.ecr.us-west-2.amazonaws.com/clouddev

2. Build(?)/tag/push images

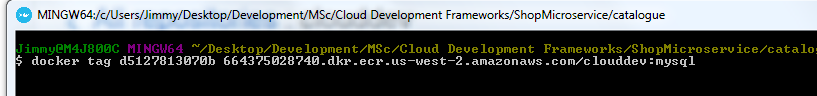
View repositories - <https://us-west-2.console.aws.amazon.com/ecs/home?region=us-west-2#/repositories>

Open Docker and use the docker login command obtained in Step 3 to authenticate with EC2.

You should get a ‘Login Succeeded’ message:



Tag your images (assuming they are pre-built on your machine) for deployment to EC2, using the repository URL from Step 2:

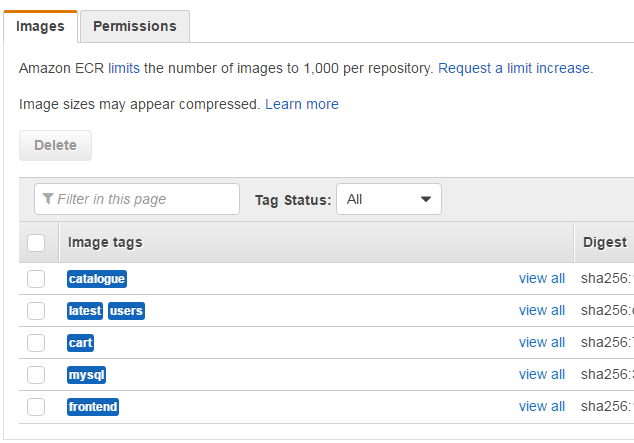


Push the image to the Repository we created in Step 2:



Repeat steps for all images.

You should now see the image in your ECR repository:



That’s the Docker images deployed.

Next to create a Task Definition.

Ref: <http://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definitions.html>

“*A task definition is like a blueprint for your application. In this step, you will specify a task definition so Amazon ECS knows which Docker image to use for containers, how many containers to use in the task, and the resource allocation for each container.*”

Ref: <http://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-task-definition.html>

Need to complete steps here - <http://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-iam-roles.html#create_task_iam_policy_and_role>

Using Policy JSON from here:

<http://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_IAM_role.html>

