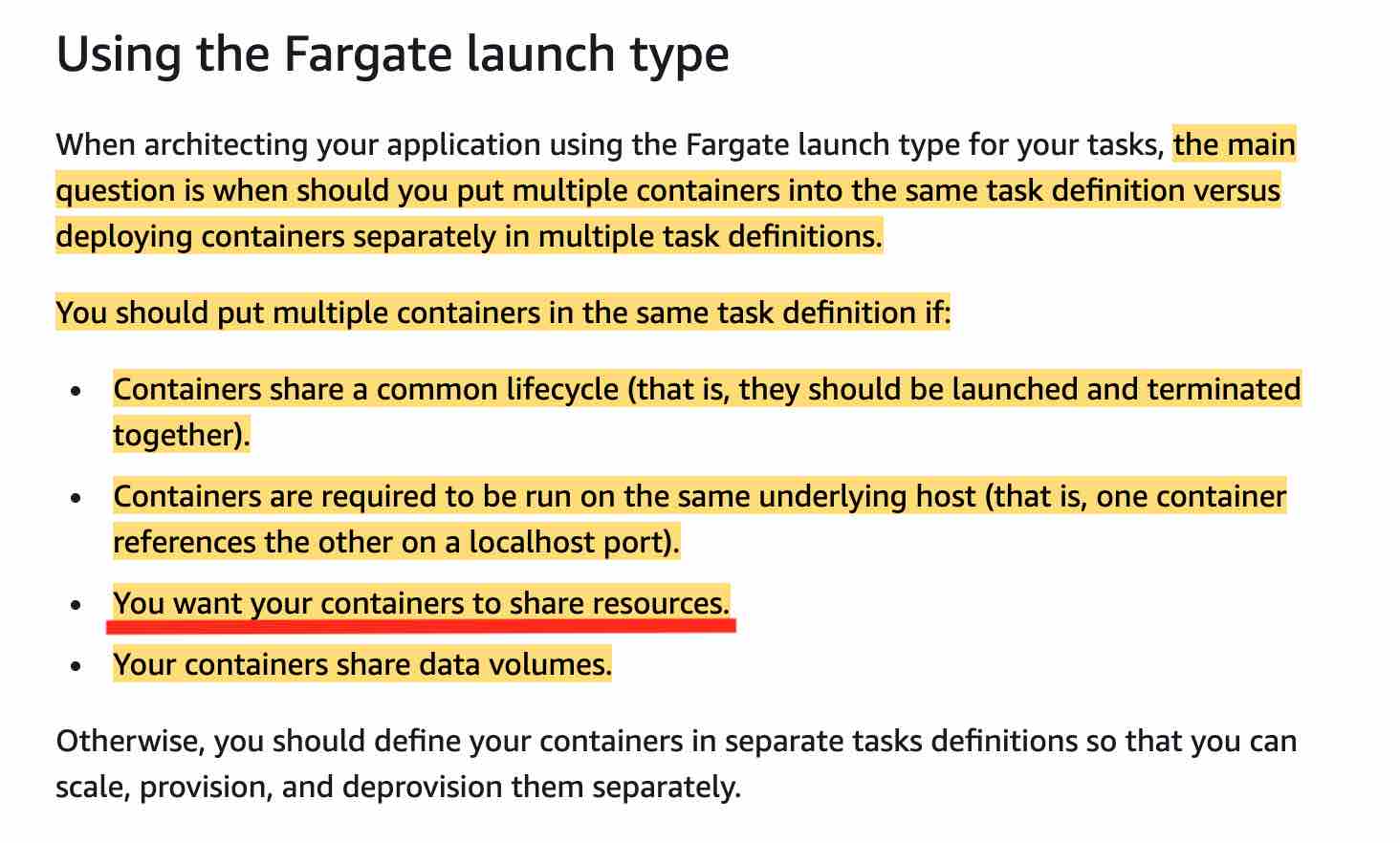
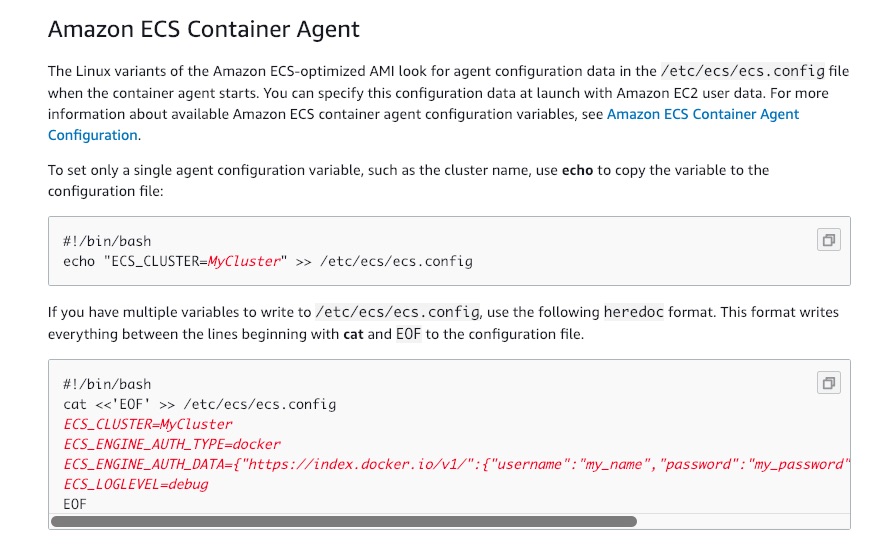
ECS

**You terminated the container instance while it was in STOPPED state, that lead to this synchronization issues** - If you terminate a container instance while it is in the STOPPED state, that container instance isn't automatically removed from the cluster. You will need to deregister your container instance in the STOPPED state by using the Amazon ECS console or AWS Command Line Interface. Once deregistered, the container instance will no longer appear as a resource in your Amazon ECS cluster.

**You terminated the container instance while it was in RUNNING state, that lead to this synchronization issues** - This is an incorrect statement. If you terminate a container instance in the RUNNING state, that container instance is automatically removed, or deregistered, from the cluster.

If you terminate a container instance in the RUNNING state, that container instance is automatically removed, or deregistered, from the cluster.





Amazon ECR users require permission to call ecr:GetAuthorizationToken before they can authenticate to a registry and push or pull any images from any Amazon ECR repository.

Q: How is Amazon ECS different from AWS Elastic Beanstalk?

AWS Elastic Beanstalk is an application management platform that helps customers easily deploy and scale web applications and services. It keeps the provisioning of building blocks (e.g., EC2, RDS, Elastic Load Balancing, Auto Scaling, CloudWatch), deployment of applications, and health monitoring abstracted from the user so they can just focus on writing code. You simply specify which container images are to be deployed, the CPU and memory requirements, the port mappings, and the container links.

Elastic Beanstalk will automatically handle all the details such as provisioning an Amazon ECS cluster, balancing load, auto-scaling, monitoring, and placing your containers across your cluster. Elastic Beanstalk is ideal if you want to leverage the benefits of containers but just want the simplicity of deploying applications from development to production by uploading a container image. You can work with Amazon ECS directly if you want more fine-grained control for custom application architectures.

Q: How is Amazon ECS different from AWS Lambda?

Amazon ECS is a highly scalable Docker container management service that allows you to run and manage distributed applications that run in Docker containers. AWS Lambda is an event-driven task compute service that runs your code in response to “events” such as changes in data, website clicks, or messages from other AWS services without you having to manage any compute infrastructure.

Q: I want to launch containers. Why do I have to launch tasks?

Docker encourages you to split your applications up into their individual components, and Elastic Container Service is optimized for this pattern. Tasks allow you to define a set of containers that you would like to be placed together (or part of the same placement decision), their properties, and how they may be linked. Tasks include all the information that Amazon ECS needs to make the placement decision. To launch a single container, your task Definition should only include one container definition.

Q: Does Amazon ECS support applications and services?

Yes. The Amazon ECS Service scheduler can manage long-running applications and services. The service scheduler helps you maintain application availability and allows you to scale your containers up or down to meet your application's capacity requirements. The service scheduler allows you to distribute traffic across your containers using Elastic Load Balancing. Amazon ECS will automatically register and deregister your containers from the associated load balancer.

The service scheduler will also automatically recover containers that become unhealthy (fail ELB health checks) or stop running to ensure you have the desired number of healthy containers supporting your application.

You can scale your application up and down by changing the number of containers you want the service to run. You can update your application by changing its definition or using a new image. The scheduler will automatically start new containers using the new definition and stop containers running the previous version (waiting for the ELB connections to drain if ELB is used).

Q: How does Amazon ECS isolate containers belonging to different customers?

Amazon ECS schedules containers for execution on customer-controlled Amazon EC2 instances or with AWS Fargate and builds on the same isolation controls and compliance that are available for EC2 customers. Your compute instances are located in a Virtual Private Cloud (VPC) with an IP range that you specify. You decide which instances are exposed to the Internet and which remain private.

* Your EC2 instances use an IAM role to access the ECS service.
* Your ECS tasks use an IAM role to access services and resources.
* Security Groups and networks ACLs allow you to control inbound and outbound network access to and from your instances.
* You can connect your existing IT infrastructure to resources in your VPC using industry-standard encrypted IPsec VPN connections.
* You can provision your EC2 resources as Dedicated Instances. Dedicated Instances are Amazon EC2 Instances that run on hardware dedicated to a single customer for additional isolation.

Q: Can I apply additional security configuration and isolation frameworks to my container instances?

Yes. As an Amazon EC2 customer, you have root access to the operating system of your container instances, enabling you to take ownership of the operating system’s security settings as well as load and configure additional software components for security capabilities such as monitoring, patch management, log management and host intrusion detection.

Q: How do I configure IAM roles for ECS tasks?

You first need to create an IAM role for your task, using the 'Amazon EC2 Container Service Task Role’ service role and attaching a policy with the required permissions. When you create a new task definition or a task definition revision you can then specify a role by selecting it form the ’Task Role’ drop-down or using the ‘taskRoleArn’ filed in the JSON format.

Q: What’s the best way to manage my repositories and images?  
Amazon ECR provides a [command line interface and APIs](http://aws.amazon.com/documentation/ecr/) to create, monitor, and delete repositories and set repository permissions. You can perform the same actions in the Amazon ECR Management Console, which can be accessed via the “Repositories” section of the Amazon ECR Console. Amazon ECR also integrates with the Docker CLI allowing you to push, pull, and tag images on your development machine.

Q: How do I publicly share an image using ECR?

You publish an image to the ECR public gallery by signing into your AWS account and pushing to a public repository you create. You are assigned a unique alias per account to use in image URLs that identifies all public images that you publish.