



Fargate Technical Questions

I capture the questions from the video and used [this](#) to convert image to text.

From there i handled the questions with a good prompt to GPT to act as an experienced AWS employee and answer the questions below. Enjoy the boost.

- **How many threads or hyperthreads do you get per vCPUs?**

You get 2 threads or hyperthreads per vCPU.

- **How do you optimize your web-server when you have vCPUs and you are using containers?**

To optimize your web server when using containers with vCPUs, you can consider the following approaches:

Efficiently allocate resources to your containers based on their requirements.

Implement horizontal scaling by running multiple container instances to distribute the load.

Use container orchestration platforms like Kubernetes or Amazon ECS to manage and scale your containers automatically.

Optimize the application code and container configuration for better performance.

Monitor resource usage and tune your web server accordingly.

Additional Strategies: Resource allocation, Horizontal scaling, Optimize container image, Caching and content delivery, Load balancing, Performance monitoring and tuning, Container-specific optimizations.

- **How many processes for a web-server should be running in a container?**

Generally, it is recommended to have one process per container for a web server.

- **Many online examples in the AWS documentation and various locations don't appear to leverage more complex web-services and instead use the simple or development servers, is this due to the documentation not showing a production use case or does the cloud provide layers of benefits (scaling, security, monitoring, logging) and features so your container can run a simple web server?**

The online examples in AWS documentation and other locations often use simple or development servers to demonstrate the basic concepts and functionality. However, in production environments, the cloud provides additional layers of benefits and features that enable running more complex web services. These benefits include scaling capabilities, security features, monitoring tools, and

logging solutions. While simple web servers may be used in the examples, the cloud infrastructure can support running more sophisticated and scalable web services.

- **Is there a use-case for using bridge networking mode?**

Yes, there are use-cases for using bridge networking mode in container environments. Bridge networking allows containers to communicate with each other on the same host without exposing their ports directly to the host network. It provides network isolation and allows running multiple containers on the same host with their own internal IP addresses. Bridge networking is commonly used for multi-container applications where containers need to communicate with each other within the same host.

- **What overhead cost does Envoy containers introduce?**

When using Fargate in Amazon ECS with App Mesh, each Envoy container typically introduces an overhead cost of 512 MB of memory.

- **Is Service Connect the recommended way over manually configuring App Mesh and Cloud Map?**

Yes, Service Connect is the recommended way for configuring App Mesh and Cloud Map in Amazon ECS. Service Connect simplifies the configuration process by providing a managed service mesh implementation that integrates with Cloud Map for service discovery. It offers an easier and more streamlined way to set up and manage the service mesh compared to manual configuration.

- **Is VPC Lattice something that could replace Service Connect, would it result in a headless service mesh?**

VPC Lattice is an internal project at Amazon that is not publicly available or documented. Therefore, its specific capabilities and features are not known. As of now, Service Connect is the recommended way for configuring a service mesh in Amazon ECS.

- **What is the purpose of the DNS option when mapping a service using Service Connect?**

The DNS option when mapping a service using Service Connect allows you to associate a DNS name with the service in Cloud Map. This DNS name can then be used to access the service within the service mesh. It provides a convenient and human-readable way to reference the service instead of using direct IP addresses.

- **When does it make sense not to use a service mesh?**

Service meshes are typically beneficial for complex microservices architectures where service-to-service communication needs to be managed, secured, and monitored.

- **Do we even need App Mesh (even through Service Connect), why isn't Cloud Map enough? What is the use case of just one or using both together?**

Cloud Map provides service discovery capabilities, allowing you to register and discover services within your environment. However, App Mesh goes beyond service discovery and provides additional features for managing and securing service-to-service communication in a microservices architecture. It offers advanced traffic management, observability, and security features like traffic routing, load balancing, metrics, and encryption.

You can use Cloud Map alone for basic service discovery needs, especially in simpler architectures. However, when you require more granular control over traffic management and enhanced observability, incorporating App Mesh becomes beneficial. By using both Cloud Map and App Mesh together, you can achieve comprehensive service discovery, advanced traffic control, and observability within your microservices architecture.

- **For future technical pathing, can Fargate serve as a stepping stone or a pivot point to using Kubernetes since we can run Fargate in EKS?**

Yes, Fargate can serve as a stepping stone or a pivot point towards using Kubernetes. With Amazon Elastic Kubernetes Service (EKS) and Fargate integration, you can run Fargate pods within an EKS cluster. This allows you to leverage the benefits of Fargate, such as serverless container execution, while gradually transitioning towards a managed Kubernetes environment.

you can simplify the management and operations of your container workloads. As your requirements evolve, you can gradually introduce Kubernetes-specific features and tools by migrating your Fargate workloads to EKS. This provides a scalable and flexible path for transitioning from Fargate to Kubernetes, based on your organization's needs.

- **Is Copilot intended to get people up and running, but you should really maintain your own CFN, CDK, or Terraform IaC?**

Copilot is an open-source command-line tool developed by AWS to simplify the deployment and management of containerized applications on AWS. It is designed to streamline the process of setting up and deploying containerized applications on Amazon ECS and AWS Fargate.

While Copilot provides convenience and abstraction for common deployment tasks, it is not intended to replace more robust infrastructure-as-code (IaC) tools like AWS CloudFormation (CFN), AWS Cloud Development Kit (CDK), or Terraform. These IaC tools offer more flexibility and control for managing complex infrastructure and deploying applications across multiple environments.

Copilot can serve as a quick and easy way to get started with containerized deployments, especially for simpler use cases.

However, for advanced scenarios or when you require fine-grained control over your infrastructure, maintaining your own CFN, CDK, or Terraform IaC code may be more appropriate.

- **Is there any option besides using Service Connect or App Mesh with ECS Fargate to get a more flexible service mesh, since as far as I understand, you can't configure the envoys managed by App Mesh.**

When using ECS Fargate, Service Connect and App Mesh are the recommended options for implementing a service mesh. While you can't directly configure the envoys managed by App Mesh in ECS Fargate, App Mesh allows you to define the traffic routing and other configuration through App Mesh resources such as virtual services, virtual nodes, and routes.

If you require a more flexible service mesh implementation with direct control over the Envoy proxies, you may consider using a different container orchestration platform, such as Amazon EKS (Elastic Kubernetes Service). With EKS, you have the flexibility to manage and configure the Envoy proxies directly within Kubernetes using tools like Istio, Linkerd, or Consul Connect, which provide advanced service mesh capabilities

- **Is it okay to be using an ALB with Service Connect instead of an NLB? When using App Mesh and Cloud Map, do you have to use an NLB?**

Yes, it is okay to use an Application Load Balancer (ALB) with Service Connect instead of a Network Load Balancer (NLB). Both ALB and NLB can be used as the load balancer for your services when utilizing Service Connect.

When using App Mesh and Cloud Map together, you have the flexibility to choose between ALB and NLB based on your specific requirements. ALB can be used for HTTP and HTTPS traffic, providing advanced features like path-based routing, content-based routing, and integration with AWS services. On the other hand, NLB is generally recommended for TCP/UDP traffic or when you require static IP addresses for your services.

The choice between ALB and NLB depends on your application's needs, the protocol you are using, and the specific features required for load balancing and traffic management.

- **What use-case would we need to have ECS Fargate with attached storage like EFS?**

ECS Fargate with attached storage like Amazon Elastic File System (EFS) can be useful in scenarios where you need shared storage across multiple containers or require persistent storage for your containerized applications.

Some common use-cases for ECS Fargate with EFS include:

- Sharing configuration files or other shared data between containers running in different Fargate tasks.
- Persisting application data that needs to be accessible across multiple Fargate tasks or even across task restarts.
- Supporting stateful applications that require a consistent file system, such as databases or content management systems.
- Enabling file-based caching or storing application assets that need to be shared among multiple instances of your containers.

EFS provides scalable and highly available network file storage that can be easily mounted to Fargate tasks, allowing you to achieve persistent and shared storage capabilities in your containerized applications.

- **How easy is it to enable Container Insights? Is it expensive to turn on?**

Enabling Container Insights, which provides monitoring and performance metrics for your containers and containerized applications, is relatively straightforward. To enable Container Insights in Amazon ECS or Amazon EKS, you can follow the AWS documentation, which provides step-by-step instructions.

Regarding cost, enabling Container Insights itself does not incur additional charges. However, you should consider the costs associated with the storage and analysis of the collected monitoring data. The data is stored in Amazon CloudWatch Logs and Amazon CloudWatch Metrics, and the pricing for these services applies.

It's important to review the pricing details of CloudWatch Logs and Metrics to understand the cost implications based on the volume of logs and metrics generated by your containerized applications. By default, CloudWatch retains logs for 30 days, but you can adjust the retention period to manage costs effectively.

In summary, while enabling Container Insights is easy, it's important to consider the potential cost implications based on the storage and analysis of monitoring data in CloudWatch Logs and Metrics.