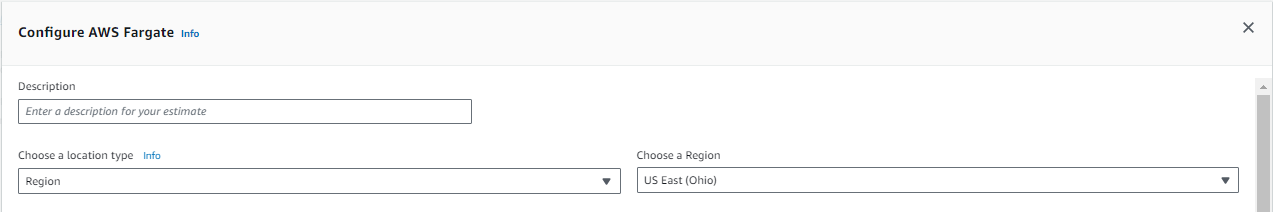
**What is AWS Fargate?**

AWS Fargate is a serverless compute engine provided by Amazon Web Services (AWS) that enables you to run containers without having to manage the underlying infrastructure. Containers are a lightweight and portable way to package and deploy applications and their dependencies. With Fargate, you can focus solely on building, deploying, and scaling your containerized applications without needing to worry about server provisioning, scaling, or management.

**Here are some key features and concepts of AWS Fargate:**

* **Serverless:** Fargate abstracts away the need to manage virtual machines or servers. You don't need to worry about infrastructure provisioning, patching, or scaling. AWS takes care of the underlying infrastructure for you.
* **Containerized Applications**: Fargate is designed to run containers. Containers are isolated environments that package an application and its dependencies, making them consistent and portable across different environments.
* **Orchestration:** Fargate integrates with container orchestration services like Amazon ECS (Elastic Container Service) and Amazon EKS (Elastic Kubernetes Service). These services allow you to define how your containers should be scheduled, deployed, and scaled.
* **Resource Allocation:** When you define a task or pod in Fargate, you specify the CPU and memory resources required by your containers. Fargate ensures that your containers have the necessary resources to run optimally.
* **Networking:** Fargate tasks have their own networking resources, including IP addresses and network interfaces. You can control networking configurations like security groups, subnets, and load balancers.
* **Scaling:** Fargate can automatically scale your tasks based on demand. You can define autoscaling policies to adjust the number of tasks based on CPU or memory utilization.
* **Isolation:** Each Fargate task runs in its own isolated environment, providing security and isolation between different tasks.
* **Pay-as-You-Go:** With Fargate, you pay for the compute resources you use on a per-task basis. This allows you to align your costs with actual usage.
* **Task Definitions:** You define your application's requirements in a task definition, including the Docker image, resource requirements, networking, and other settings.
* Integration with Ecosystem: Fargate integrates with other AWS services, such as Amazon CloudWatch for monitoring, AWS Identity and Access Management (IAM) for access control, and more.

Overall, AWS Fargate simplifies the process of deploying and managing containerized applications by abstracting away the complexity of infrastructure management. It allows developers to focus on writing code and deploying applications while AWS takes care of the operational aspects.

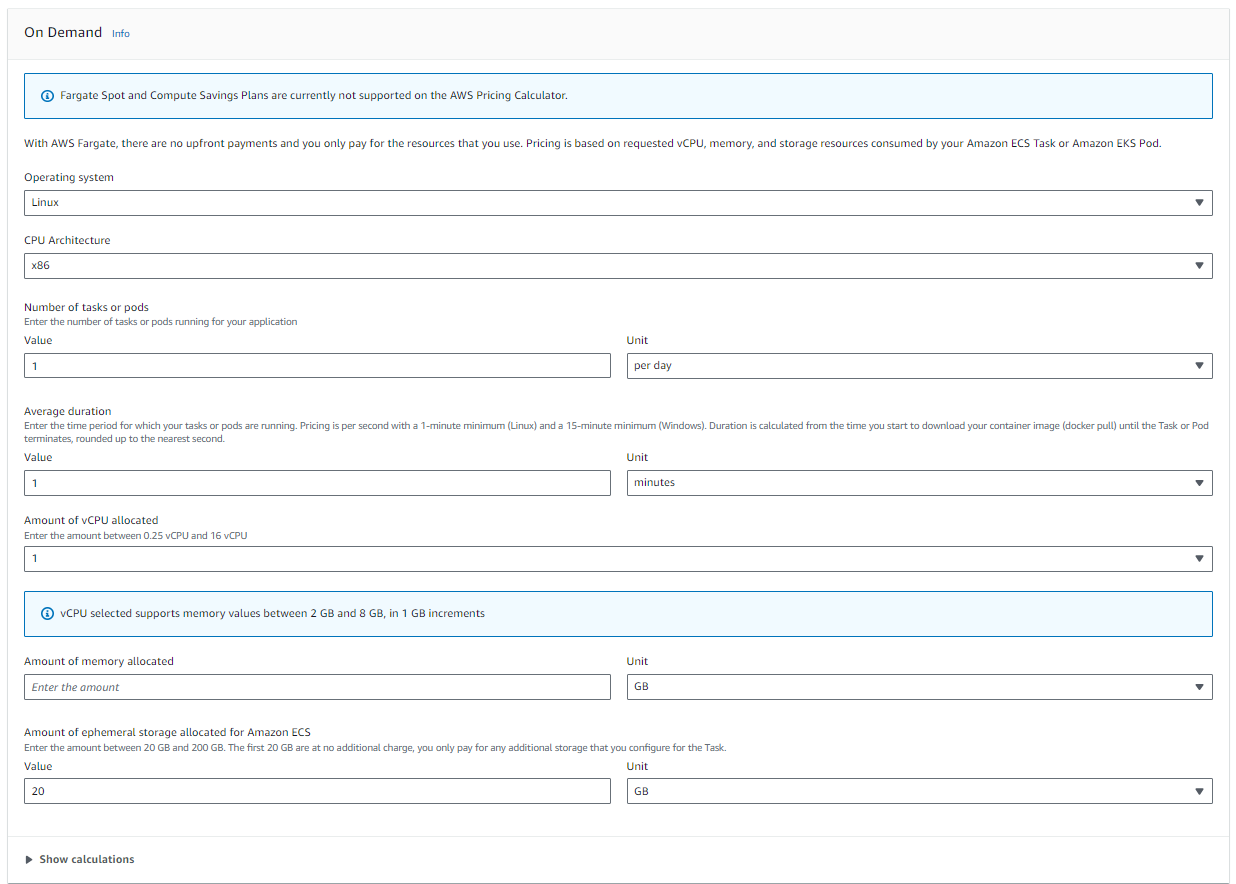
****

* **Description:** Name of Fargate
* **Choose a Region:** Select Region of the Fargate
* **Fargate Region:**

AWS Fargate is available in multiple AWS regions around the world. AWS regions are physical locations where AWS data centers are located, and each region consists of multiple Availability Zones, which are isolated data centers within a region. Fargate is designed to be available in multiple regions to provide high availability and fault tolerance for your containerized applications.

When you use AWS Fargate, you can choose the region where you want to deploy your containerized applications. The choice of region can impact factors such as latency, data sovereignty, and compliance requirements. Different AWS services and features, including Fargate, may have availability and features that vary by region.

It's important to note that AWS continues to expand its services and regions, so there may have been developments or changes related to AWS Fargate's availability in specific regions.

****

**On Demand**

With AWS Fargate, there are no upfront payments and you only pay for the resources that you use. Pricing is based on requested vCPU, memory, and storage resources consumed by your Amazon ECS Task or Amazon EKS Pod.

* **Operating system:** Windows, Linux
* **CPU Architecture:** x86, ARM
* **Number of tasks or pods:** Enter the number of tasks or pods running for your application:

1. **Value:** Number of tasks or pods
2. **Unit:** Second, Minute, hour, days, Months

* **Average duration:** Enter the time period for which your tasks or pods are running. Pricing is per second with a 1-minute minimum (Linux) and a 15-minute minimum (Windows). Duration is calculated from the time you start to download your container image (Docker pull) until the Task or Pod terminates, rounded up to the nearest second.

1. **Value:** from1 to 999
2. **Unit:** Second, Minute, hour, days

* **Amount of vCPU allocated:** Enter the amount between 0.25 vCPU and 16 vCPU : 0.25 to 16
* **Amount of memory allocated**:

1. **0.25** --- 0.5 GB, 1 GB, or 2 GB
2. **0.5** --- 1 GB and 4 GB, in 1 GB increments
3. **1** --- 2 GB and 8 GB, in 1 GB increments
4. **2** --- 4 GB and 16 GB, in 1 GB increments
5. **4** --- 8 GB and 30 GB, in 1 GB increments
6. **8** --- 16 GB and 60 GB, in 4 GB increments
7. **16** --- 32 GB and 120 GB, in 8 GB increments

* **Amount of ephemeral storage allocated for Amazon ECS:** Enter the amount between 20 GB and 200 GB. The first 20 GB are at no additional charge, you only pay for any additional storage that you configure for the Task.

1. **Value**: 20 to 200
2. **Unit**: GB

