

		AWS COMPUTE SERVICES						
FACTORS		EC2	ECS on EC2	ECS on Fargate	AWS Batch	EKS on EC2	EKS on Fargate	Lambda
Simplicity (1=complex - 5=simple)		1	2	3	3	1	2	4
		EC2 is highly configurable and there are numerous non-trivial options	Containerisation gives some simplicity but EC2 still has to be configured as well as ECS itself	No instances but requires cluster, task definition, service configuration	Uses ECS on EC2 but with an option to manage the EC2 cluster	EC2 complexity remains with the added broad complexity of Kubernetes	Fargate removes the EC2/OS configuration but the broad complexity of Kubernetes should be considered	Configuration, failure handling and awareness of constraints
Cost of Maintenance (1=low - 5=high)		5	4	3	3.5	5	4	1.5
		OS, network, security, more complex deployment models	ECS Cluster maintenance overhead is non-trivial	ECS Cluster maintenance overhead is non-trivial	Limited visibility, requires custom metrics and insight	Significant complexity comes with Kubernetes, OS, network, security	Significant complexity remains with Kubernetes	Executions, failures, performance, events and cost all need monitoring and ongoing adjustment
Scalability	Unit of Scale	Instance	Instance + Task	Task	Instance + Job	Instance + Pod	Pod	Function Invocation
	How does it scale?	Auto-scaling	Auto-scaling	Auto-scaling	Batch queues and job scheduler with ECS task scheduler	K8S Cluster Autoscaler + Scheduler	K8S Cluster Autoscaler + Scheduler	On demand driven by events
	Scaling Speed	Instance type and configuration dependent	60 tasks per minute*	~20 tasks per minute*	~60 jobs per minute	~500 pods per minute	~20 pods per minute*	Immediate burst to 3000 (500/1000 in some regions) 500 / minute thereafter
	Scaling Limits	Limited by vCPU quotas	<ul style="list-style-type: none"> Limited by vCPU quotas 2000 tasks per cluster soft limit 	100 task soft limit	Limited by vCPU quotas	Limited by vCPU quotas	<ul style="list-style-type: none"> 500 pod soft limit 	<ul style="list-style-type: none"> Account level concurrency quota (default 1000) SQS events scaling limited to 60/minute!
	Constraints			<ul style="list-style-type: none"> 4 vCPUs 30 GB RAM 20 GB storage 			<ul style="list-style-type: none"> 4 vCPUs 30 GB RAM 20 GB storage 	<ul style="list-style-type: none"> 3008 MB Memory 512MB Storage 250 MB Code 15 minute timeout No GPU's
Service Integrations	VPC, EBS, SSM	ECR • AppMesh • ELB	ECR • AppMesh • ELB • Step Functions	ECR • Step Functions • Elastic Fabric Adapter	ECR • AppMesh • ELB	ECR • AppMesh • ELB • Step Functions	EFS • RDS • VPC • S3 • API Gateway • DynamoDB • Kinesis • EventBridge • Glue • Step Functions • SQS • SNS • MSK • MQ	
Unique Features	<ul style="list-style-type: none"> Huge range of instance types to suit specific workloads Spot instances Reserved Instances 		<ul style="list-style-type: none"> Fargate Spot Managed containers 	<ul style="list-style-type: none"> Job scheduler Array jobs Priority queues Job dependencies 	<ul style="list-style-type: none"> Large K8S ecosystem 	<ul style="list-style-type: none"> Large K8S ecosystem 	<ul style="list-style-type: none"> Reserved concurrency Provisioned concurrency Destinations DLQ 	

* Fargate and ECS scaling speed may be improved with limit increases through AWS Support