Amazon - ECS

ECS

- Amazon ECS enables you to inject sensitive data into your containers by storing your sensitive data in either AWS Secrets Manager secrets or AWS Systems Manager Parameter Store parameters and then referencing them in your container definition.
- This feature is supported by tasks using both the EC2 and Fargate launch types.
- Secrets can be exposed to a container in the following ways:
- - To inject sensitive data into your containers as environment variables, use the secrets container definition parameter.
- - To reference sensitive information in the log configuration of a container, use the secretOptions container definition parameter.

Service Auto Scaling

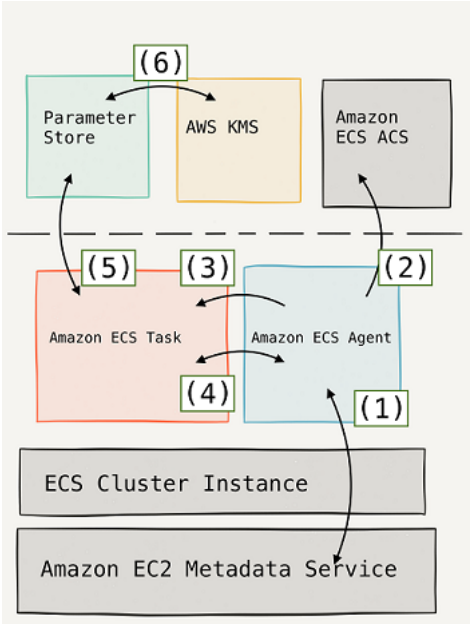
- Amazon ECS service can optionally be configured to use Service Auto Scaling to adjust its desired count up or down in response to CloudWatch alarms. Service Auto Scaling leverages the Application Auto Scaling service to provide this functionality.
- Service Auto Scaling is available in all regions that support Amazon ECS.

Reference:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-auto-scaling.html

Check out this Amazon ECS Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-elastic-container-service-amazon-ecs/



ESC Components:

- Containers and Images,
- Task Components :
  - Task Family, IAM task role, Network mode, Container Definition, Volumes, Task Placement constraints, Launch Types
- Tasks and Scheduling
  - Task
  - Scheduler - REPLICA, DAEMON
- Clusters : Placement group i.e., Logical group of resources
  - Fargate launch type : AWS manages
  - EC2 launch type : customer manages
- Services

	○ Rolling Update ○ Blue/Green with Code Deploy • Container Agent ○ Runs on each infra resource within ECS cluster ○ You can attach multiple target groups

- Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of either the Secrets Manager secret or Systems Manager Parameter Store parameter containing the sensitive data to present to the container.
- The parameter that you reference can be from a different Region than the container using it, but must be from within the same account.

RIGHT ANSWER

Use the AWS Systems Manager Parameter Store to keep the database credentials and then encrypt them using AWS KMS. Create an IAM Role for your Amazon ECS task execution role (taskRoleArn) and reference it with your task definition, which allows access to both KMS and the Parameter Store. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Systems Manager Parameter Store parameter containing the sensitive data to present to the container.

WRONG ANSWER

Use the AWS Secrets Manager to store the database credentials and then encrypt them using AWS KMS. Create a resource-based policy for your Amazon ECS task execution role (taskRoleArn) and reference it with your task definition which allows access to both KMS and AWS Secrets Manager. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Secrets Manager secret which contains the sensitive data, to present to the container is incorrect because although the use of Secrets Manager in securing sensitive data in ECS is valid, using an IAM Role is a more suitable choice over a resource-based policy for the Amazon ECS task execution role.

Amazon - ECS - Task Definition

- **Task Definition for Fargate Launch Type:**
  - Network mode = awsvpc
  - Specify Memory and CPU per task
  - Only supports 'awslogs' log driver for Log configurations and sends logs to AWS cloudwatch.
  - Default Task storage is 'ephemeral', But can be mounted to EFS
  - Only Supports Images from ECR repo / Docker-hub

- **Task Definition for EC2 Launch Type:**

- Volumes = docker / bind volumes
- Support Private repos

Amazon - ECS - Task Definition

Task definitions are split into separate parts:

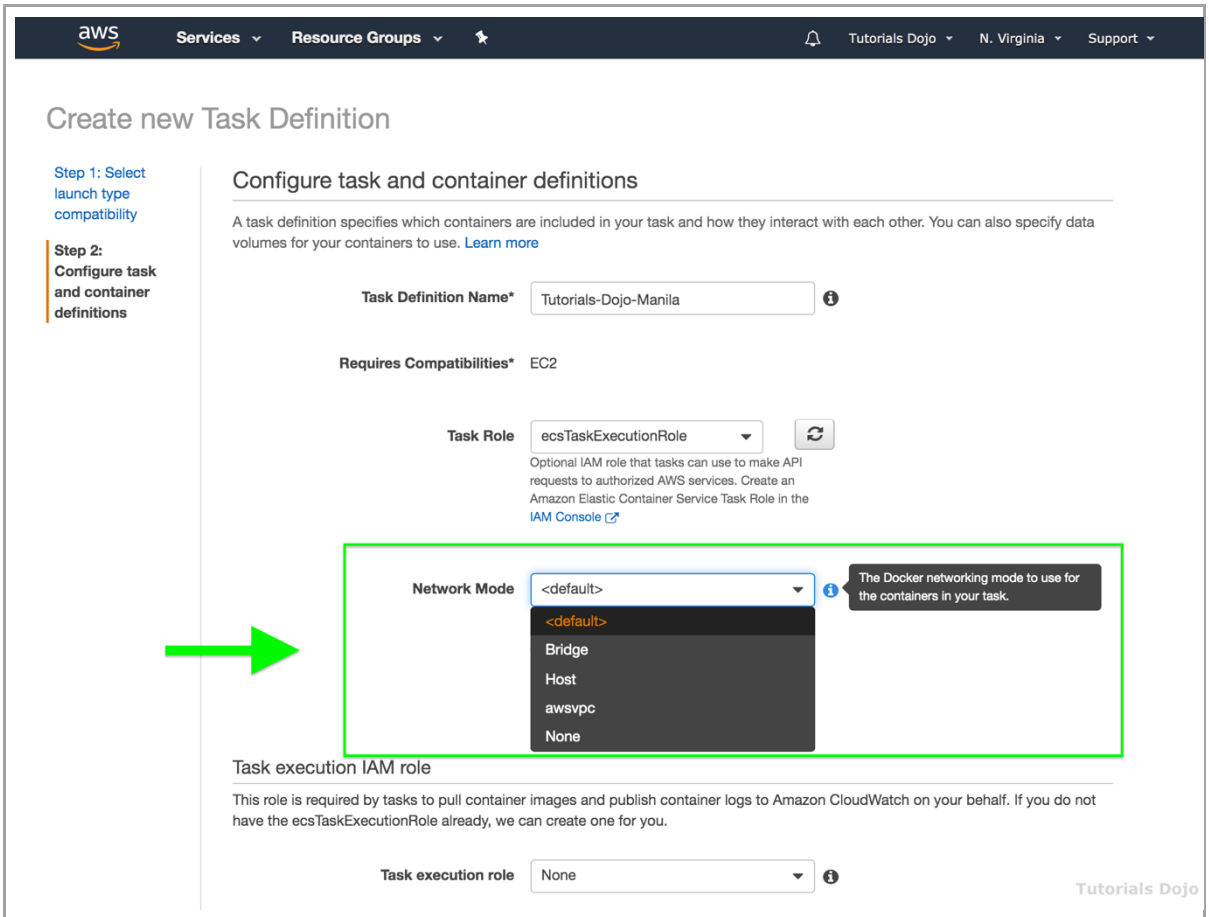
- the task family,
- the IAM task role,
- the network mode,
- container definitions,
- volumes,
- task placement constraints, and
- launch types.

The family and container definitions are required in a task definition, while task role, network mode, volumes, task placement constraints, and launch type are optional.

You can configure various Docker networking modes that will be used by containers in your ECS task. The valid values are none, bridge, awsvpc, and host. The default Docker network mode is bridge.

With IAM roles for Amazon ECS tasks, you can specify an IAM role that can be used by the containers in a task. Applications must sign their AWS API requests with AWS credentials, and this feature provides a strategy for managing credentials for your applications to use, similar to the way that Amazon EC2 instance profiles provide credentials to EC2 instances.

Instead of creating and distributing your AWS credentials to the containers or using the EC2 instance's role, you can associate an IAM role with an ECS task definition or RunTask API operation. The applications in the task's containers can then use the AWS SDK or CLI to make API requests to authorized AWS services.



If the network mode is set to **none**, the task's containers do not have external connectivity and port mappings can't be specified in the container definition.

If the network mode is **bridge**, the task utilizes Docker's built-in virtual network which runs inside each container instance.

If the network mode is **host**, the task bypasses Docker's built-in virtual network and maps container ports directly to the EC2 instance's network interface directly. In this mode, you can't run multiple instantiations of the same task on a single container instance when port mappings are used.

If the network mode is **awsvpc**, the task is allocated an elastic network interface, and you must specify a NetworkConfiguration when you create a service or run a task with the task definition. When you use this network mode in your task definitions, every task that is launched from that task definition gets its own elastic network interface (ENI) and a primary private IP address. The task networking feature simplifies container networking and gives you more control over how containerized applications communicate with each other and other services within your VPCs.

Task networking also provides greater security for your containers by allowing you to use security groups and network monitoring tools at a more granular level within your tasks. Because each task gets its own ENI, you can also take advantage of other Amazon EC2 networking features like VPC Flow Logs so that you can monitor traffic to and from your

tasks. Additionally, containers that belong to the same task can communicate over the localhost interface. A task can only have one ENI associated with it at a given time.

References:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-networking.html

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html#network_mode

Check out this Amazon ECS Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-elastic-container-service-amazon-ecs/

ECS Limits

Resource	Default Limit
Number of clusters per region, per account	1000
Number of container instances per cluster	1000
Number of services per cluster	500
Number of tasks using the EC2 launch type per service (the desired count)	1000
Number of tasks using the Fargate launch type, per region, per account	50
Number of load balancers per service	1
Number of tasks launched (count) per **run-task**	10
Number of container instances per **start-task**	10
Task definition max containers	10
Maximum layer size of an image used by a task using the Fargate launch type	4 GB
Maximum size of a shared volume used by multiple containers within a task using the Fargate launch type	4 GB
Maximum container storage for tasks using the Fargate launch type	10 GB

References:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data.html

https://aws.amazon.com/blogs/mt/the-right-way-to-store-secrets-using-parameter-store/

Check out these Amazon ECS and AWS Systems Manager Cheat Sheets:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-elastic-container-service-amazon-ecs/

https://tutorialsdojo.com/aws-cheat-sheet-aws-systems-manager/