Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use.

Creating a Task Definition.

Steps

- On main dashboard select “Task Definitions”.
- Enter a task definition name (careful when using numbers as task definitions will eventually have “revisions” that will add numbers on a 1 basisoidhjodjh)
- Network Mode: The Docker networking mode to use.
- Add a container definition: This is where most of the magic happens as this will generate the dockerfile. We can specify information such as CPU units, memory, port mappings (that will create the EXPOSE command) and most important, the Docker image. It’s important to note that the Docker image must exist in DockerHub.
- We can also use JSON to configure the container. Potential integration with custom API calls to the cluster!

Task Definitions are templates that when instantiated, will deploy the containers. This means that ECS Agent and Docker Engine will work together to pull the image from the repository and run it in the appropriate EC2 instance.

Task Definitions JSON syntax


```
{
  "requiresAttributes": [],
  "taskDefinitionArn": "arn:aws:ecs:us-west-2:520710614288:task-definition/task-node:18",
  "networkMode": "bridge",
  "status": "ACTIVE",
  "revision": 18,
  "taskRoleArn": null,
  "containerDefinitions": [
    {
      "volumesFrom": [],
      "memory": 64,
      "extraHosts": null,
      "dnsServers": null,
      "disableNetworking": null,
      "dnsSearchDomains": null,
      "portMappings": [
        {
          "hostPort": 0,
          "containerPort": 8080,
          "protocol": "tcp"
        }
      ],
      "hostname": null,
      "essential": true,
      "entryPoint": null,
      "mountPoints": [],
      "name": "image-node",
      "ulimits": null,
      "dockerSecurityOptions": null,
      "environment": [],
      "links": null,
      "workingDirectory": null,
      "readonlyRootFilesystem": null,
      "image": "acanessa/node-app",
      "command": null,
      "user": null,
      "dockerLabels": null,
      "logConfiguration": null,
      "cpu": 5,
      "privileged": null,
      "memoryReservation": null
    }
  ],
  "volumes": [],
  "family": "task-node"
}
```