**Dynatrace OneAgent Integration with AWS ECS**

**Types of Solutions**

1. **Automatic Injection EKS (Elastic Kubernetes Service)**
2. **Build Time Injection**
3. **Runtime injection**

**Illustration & Proposed solutions are given below.**

1. **Automatic Injection EKS (Elastic Kubernetes Service) –**

* With automatic injection you can manage upgrades and lifecycle
* On AWS Fargate, only the application Monitoring deployment without the CSI

driver is supported

Application-only monitoring: Automatic injection

* Use application-only injection strategy for application pods.
* don't install OneAgent pods & can't collect host metrics from Kubernetes nodes.
* can collect node and container metrics by combining it with Kubernetes Platform Monitoring

**Capabilities**

* It's engineered for Kubernetes. Dynatrace injects into pods using the Kubernetes admission controller, which injects a Dynatrace code module into application containers.
* It's flexible. You get granular control over the instrumented pods using namespaces and annotations. You can easily route pod metrics to different Dynatrace environments within the same Kubernetes cluster.
* Enables data enrichment for Kubernetes environments.

**Current limitations**

* Diagnostic files (support archives) for application pods aren't yet supported.
* Go static monitoring is partially supported.

When deployed in application-only mode, the Dynatrace code modules monitor the memory, disk, CPU, and networking of processes within the container only. Host metrics aren't monitored. Without Kubernetes Platform Monitoring, topology is limited to pods and containers.

1. **Build Time Injection –**

Docker Image needs to be built with Monitoring Agent.

Not a Flexible solution when changing the monitoring solution – Reason below

* Container Rebuild Required
* Rollout Required

**Suggesting the below Runtime Injection**

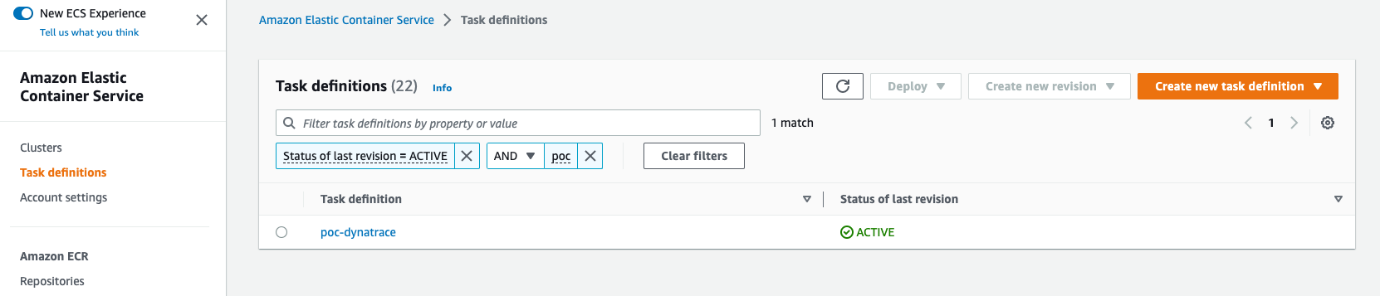
1. **Runtime injection -**

* Pulls the monitoring agent when the container starts.
* can be implemented without touching the container by loading the monitoring library during the launch phase of the application’s container.

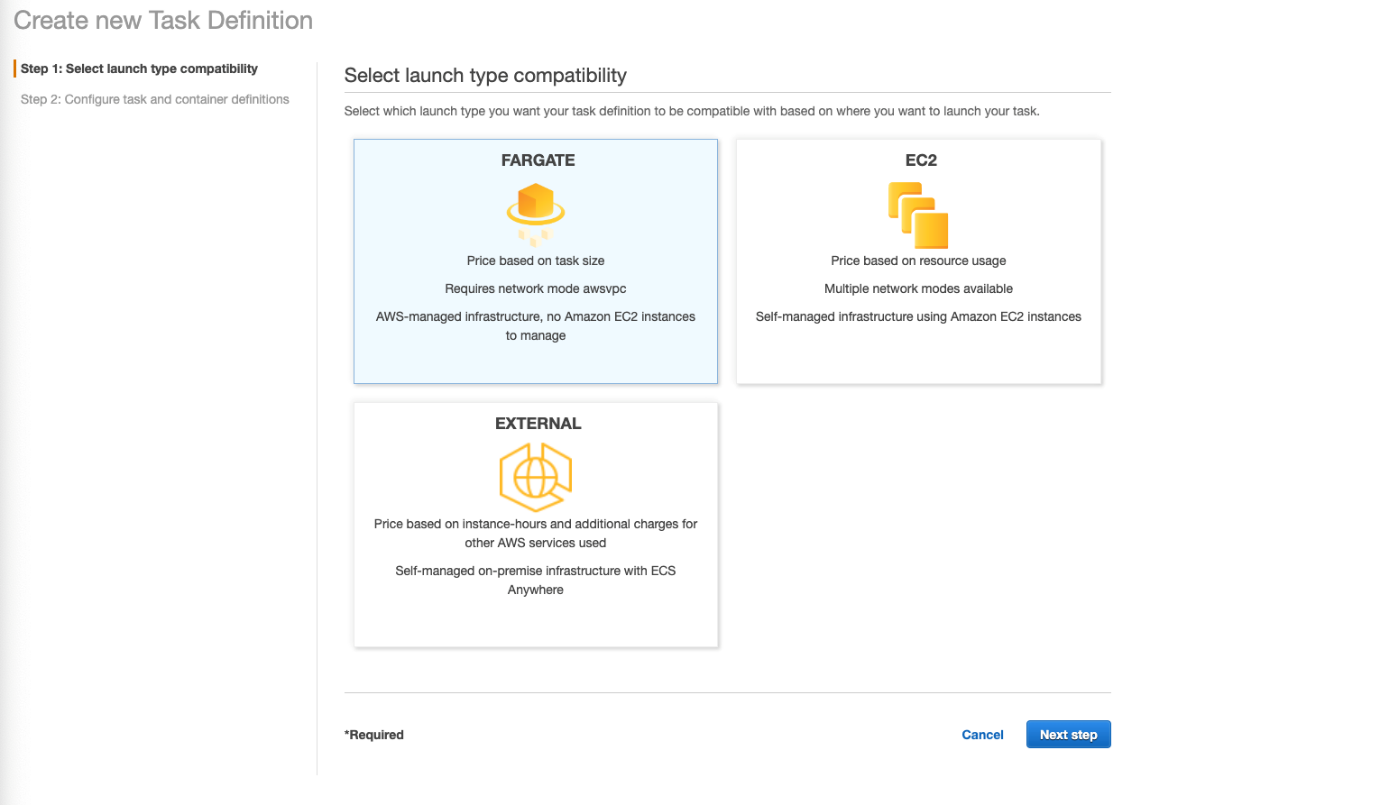
Overall, the **Runtime** **Injection** is more flexible because it decouples the installation, the configuration and the run of the Dynatrace OneAgent from the run of the application itself. Also, it allows more flexibility because you can change your monitoring solution without modifying the container build.

Steps to Install Oneagent in AWS ECS with Docker container.

In the AWS console, go to the Amazon ECS service. Under Task definitions click on “Create a new task definition”

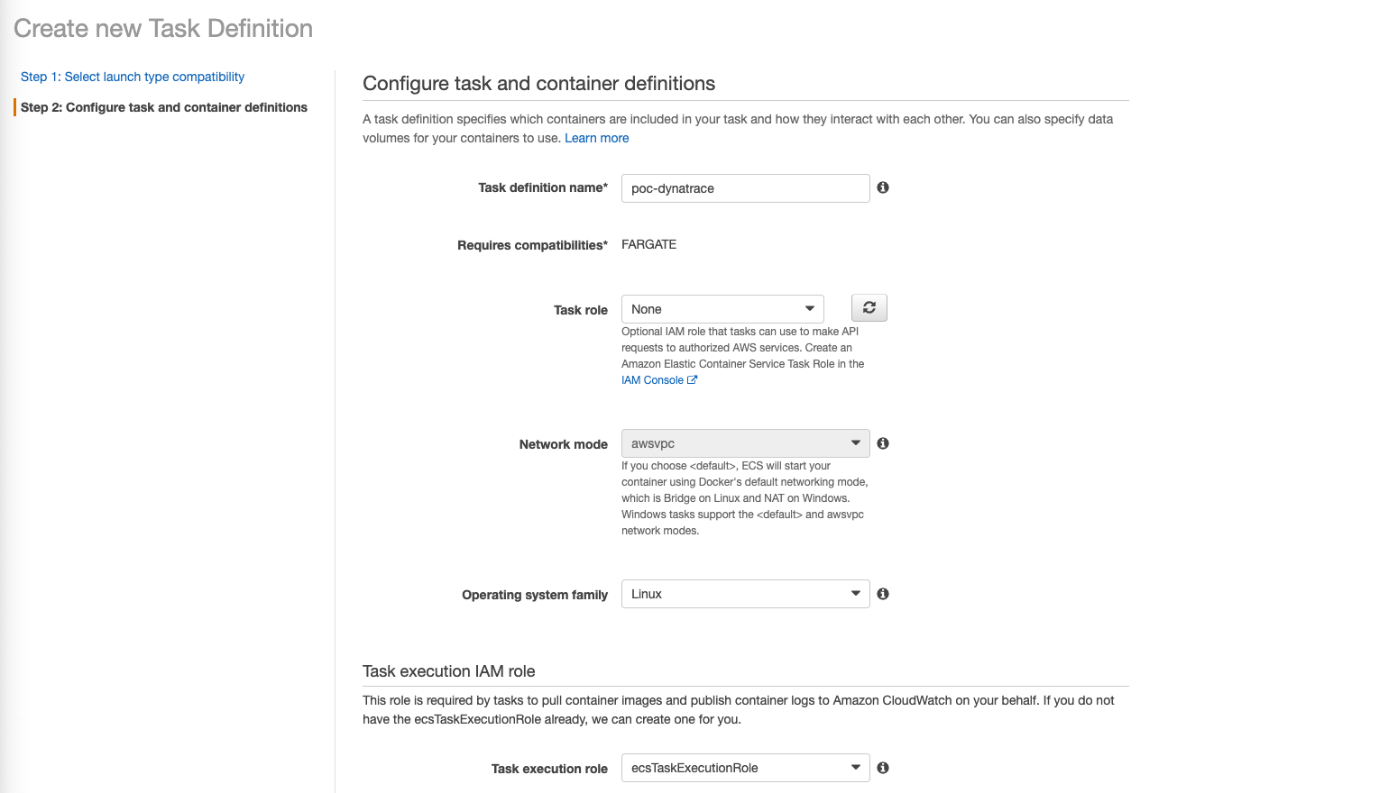


Select “Fargate” and click on “Next step”.



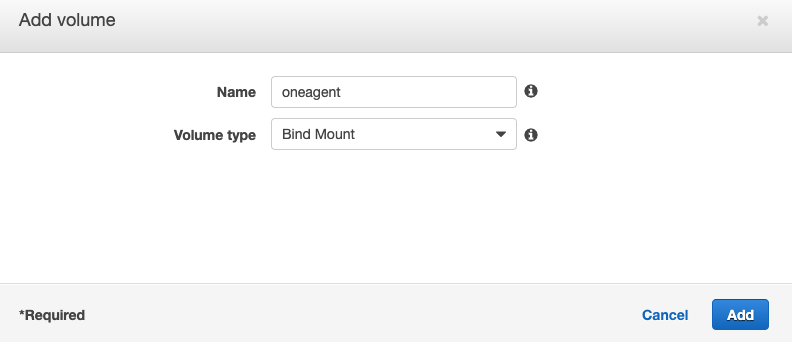
For the step “Configure task and container definitions”, enter:

* Task definition name: poc-dynatrace
* Task role: None
* Network mode: awsvpc
* Operating system family: Linux
* Task execution IAM role
  + Task execution role: ecsTaskExecutionRole
* Task size
  + Task memory (GB): 3GB
  + Task CPU (vCPU): 0.5 vCPU



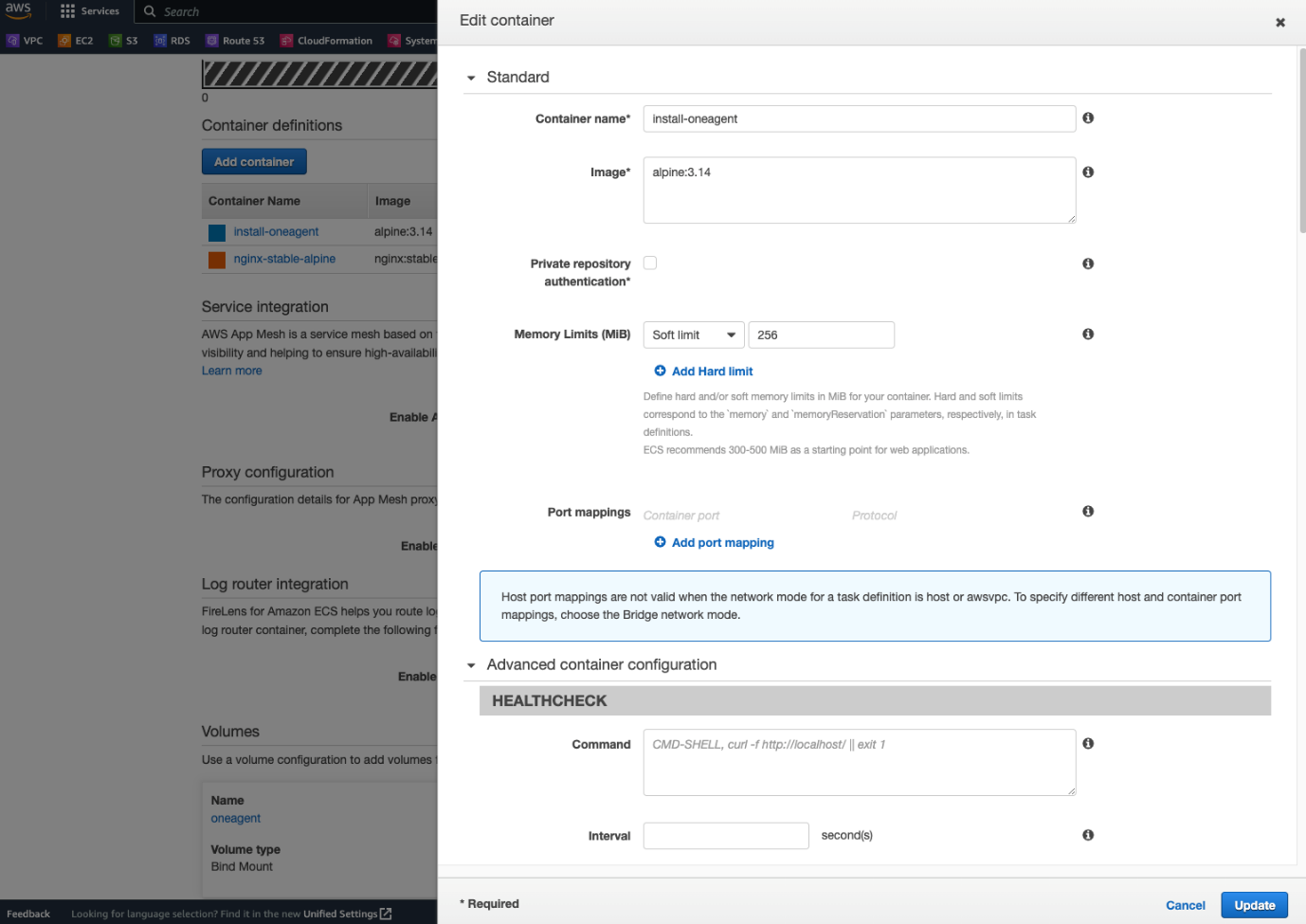
Scroll down to the “Volumes” section and add a volume. This volume is used to store monitoring libraries which will be loaded after by the application container.

* Name: Oneagent
* Volume type: Bind Mount



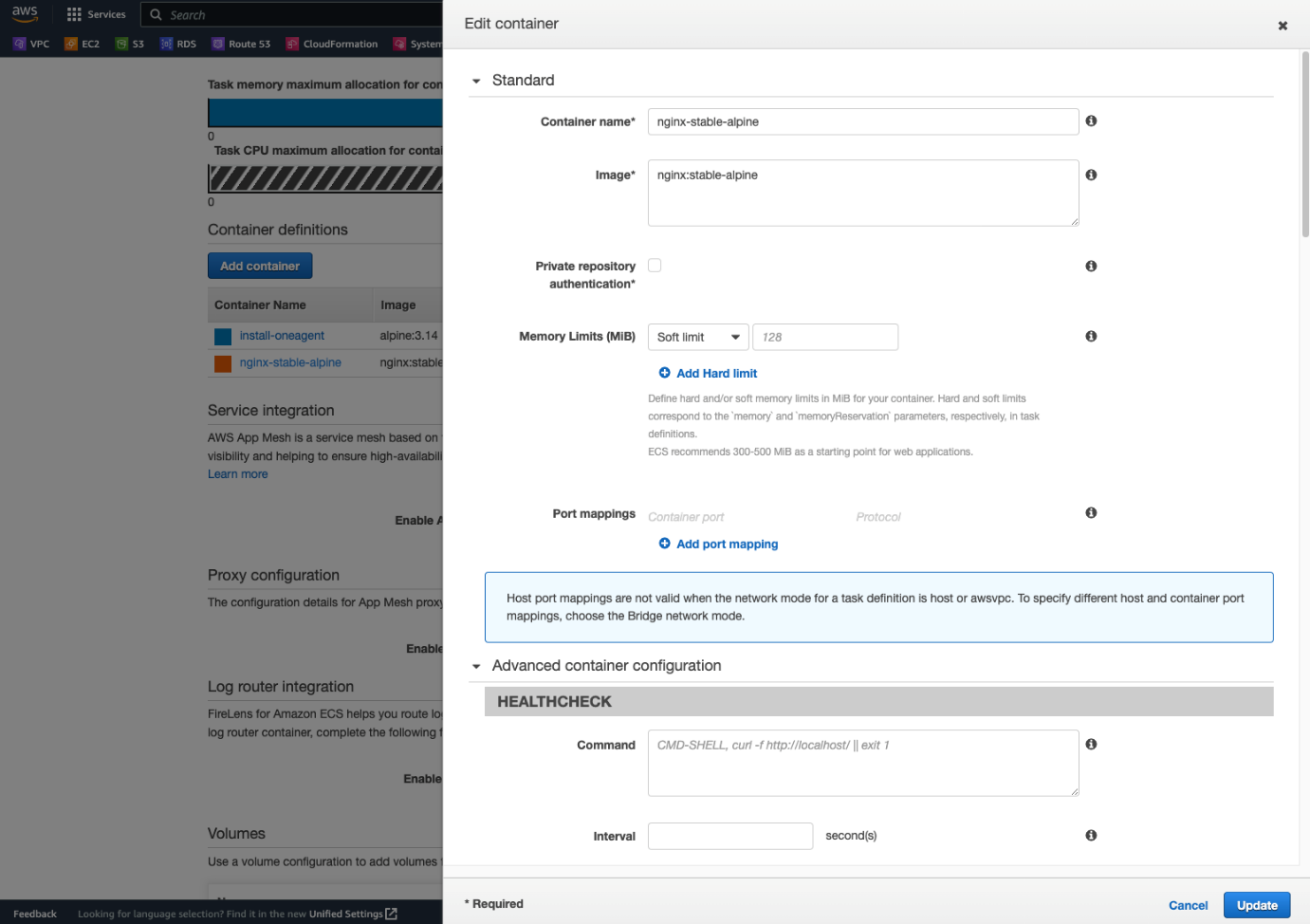
Scroll up and click on “Add container” and define the first container that will install the Dynatrace OneAgent.

* Standard
  + Container Name: install-oneagent
  + Image URI: alpine:3.14
  + Soft limit: 256
  + Remove Port mappings
* Environment
  + Entry point: /bin/sh,-c
  + Command: ARCHIVE=$(mktemp) && wget -O $ARCHIVE "$DT\_API\_URL/v1/deployment/installer/agent/unix/paas/latest?Api-Token=$DT\_PAAS\_TOKEN&$DT\_ONEAGENT\_OPTIONS" && unzip -o -d /opt/dynatrace/oneagent $ARCHIVE && rm -f $ARCHIVE
  + Environment variables
    - DT\_PAAS\_TOKEN: dt.TBD.TBD
    - DT\_API\_URL: DT\_URL\_TBD
    - DT\_ONEAGENT\_OPTION: flavor=musl&include=all
  + Uncheck Essential
* Storage and logging
  + Mount points: oneagent
  + Container path: /opt/dynatrace/onagent

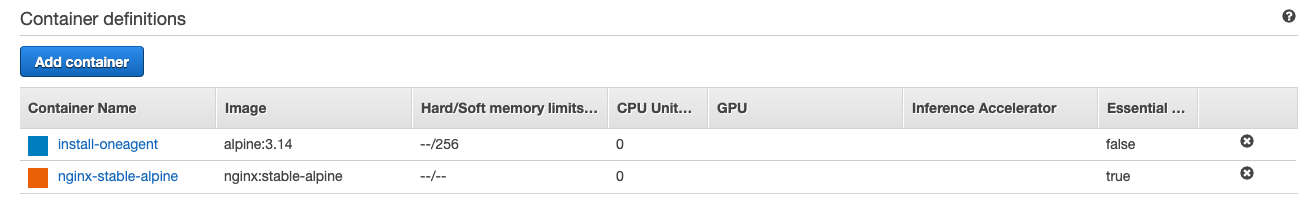


Click on “Add container” once again to add our application container.

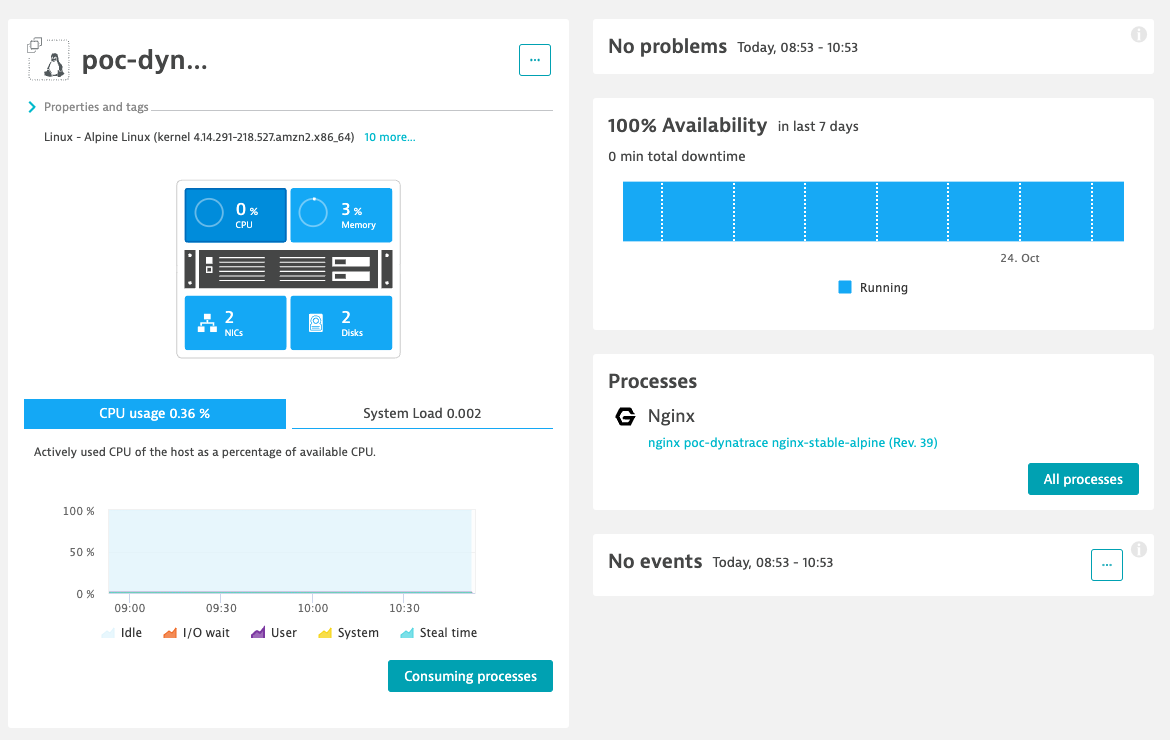
* Standard
  + Container Name: nginx-stable-alpine
  + Image URI: nginx:stable-alpine
  + Essential container: Yes
  + Add port mappings
    - 80 TCP
* Environment
  + Environment variables
    - LD\_PRELOAD: /opt/dynatrace/oneagent/agent/lib64/liboneagentproc.so
    - DT\_LOGLEVELCON: info
  + check Essential
* Startup dependency ordering
  + Container name: Install-oneagent
  + Condition: Complete
* Storage and logging
  + Mount points: oneagent
  + Container path: /opt/dynatrace/onagent



Now, you should see 2 containers defined.



Click on “Create”. Once your task definition is created, you can launch your container on your cluster, and you should see the container appear in the Dynatrace console.



You can get an overview of the list of all built-in metrics available in the Dynatrace console.