

ISSUES - PROBLEMS - MISCELLANEOUS

Extra tasks? When deploying a new revision of a task you might see more running tasks than the desired count.

SOLVED

See Mix / Max Healthy Percent

EBL - Health Check Fails. When scaling up the number of containers received Error 502 on the browser.

SOLVED

In the Events tab ECS states the reason as "Health checks failed". A quick look at the registered EC2 instances and ports in the Load Balancer shows that containers are being instantiated but exit prematurely for some reason. The ELB could not reach the container's dynamic port. A quick check on the container reveals that the configuration of the task-definition is making the container fail. Memory resources were not sufficient for the container. **Perfect opportunity to test the task-definition revision feature!** Updated the service configuration using revision 7, the **last known working configuration**. Immediately **system stability started to improve**. Events reported "service-1 has reached a steady state". "Unhealthy status" on the ports associated to the EC2 instances begun to "Drain"

Different revisions running at the same time. When updating a service with a new task-definition revision, sometimes old revisions are still running. Possibly they have a time out from visible behavior. More research needed.

SOLVED

See Min Max Healthy Value.

EC2 Instances unable to join the cluster

SOLVED

EC2 instances need ECS agent and Docker agent running.

EC2 instances still won't join cluster

SOLVED

Before launching an EC2 Instance to use in the cluster, create an IAM Role "AmazonEC2ContainerServiceforEC2Role" to allow it's use by ECS.

EC2 instances stop communicating with Cluster after changing instance type.

SOLVED

[ERROR] Could not register module="api client" err="ClientException: Container instance type changes are not supported. Container instance 86db3e95-ceeb-4ca2-ab66-fe371e77fd6a was previously registered as t2.nano.

But, why? Is this definitive?

Unhealthy ports in ELB take “too long” to drain when service is back to stable status.

Why? Check configuration?

Reduce health check timing?

SOLVED

Problem was in faulty configuration of task definition resulting in containers being created and crashing constantly.

When powering off EC2 instance, the other node is taking up the load. But when instance is up again, won't take the load back.

Configuration?

Expected behavior?

Response time stopping containers is not as fast as desired.

Docker images too big? (not likely)

EC2 instances not powerful enough? (possible)

ELB waiting for connections to be drained?

Configuration issue?

UPDATE:

docker ps -a reveals that container exited correctly (Exited (137) 6 minutes ago) but docker ps does not show this information.

UPDATE 2:

2016-09-08 15:59:25 -0400 service [service-1](#) has reached a steady state.

2016-09-08 15:59:15 -0400 service [service-1](#) has stopped 1 running tasks: task [1bbe9fb1-238c-4713-948a-f5f73a9db863](#).

2016-09-08 15:54:04 -0400 service [service-1](#) has begun draining connections on 1 tasks.

2016-09-08 15:54:04 -0400 service [service-1](#) deregistered 1 targets in target-group [ecs-cluster-target](#)

UPDATE 3:

Starting containers is fast. Two tests performed:

- Starting four containers one at the time: averages 6 seconds.
- Starting four containers at the same time: average 24 seconds.

Stopping too slow! Needs to be figured out! **CRITICAL**

UPDATE 4:

Container timeout? <https://github.com/aws/amazon-ecs-agent/issues/92>

Tune timeout? <https://github.com/aws/amazon-ecs-agent/issues/126>

When applying a new task-definition to the service, the Deployments tab will show as PRIMARY the new tasks being deployed and as ACTIVE the tasks already running. (*Why? More research needed.*)

UPDATE: This has to do with the value of **Minimum healthy percent!** This is KEY to avoid losing service when deploying a new revision of the task.

If the service stops all tasks to replace them with the new one, service would be momentarily down, that's why ECS will keep X amount of tasks running while deploying the new ones.

RESEARCH: This means two potentially different versions of the container might be running at the same time, and this can be a possible problem due to difference in code. So, MaxH% means how many tasks ECS will stop during a new deployment and MaxH% means how many tasks will ECS deploy before stopping the old tasks.

Example: 200% of 4 desired tasks = 4 new tasks. (*Weird. Double Check.*)