AWS USER GROUP PUNE

“For the community, By the community!”

aws COMMUNITY DAY

PUNE

23rd August 2025

Annual Edition

COMMUNITY DAY
PUNE
23rd August 2025
Annual Edition

USER GROUP
PUNE

Why Your ECS Tasks Won't Launch: Mastering Task Placement Strategies

Avinash Shashikant Dalvi

Senior Staff Engineer, Nushift Technologies

Full stack developer by profession and heart

AWS CBs, AWS BLR UG Leader

How many of you have had ECS tasks stuck in PENDING?

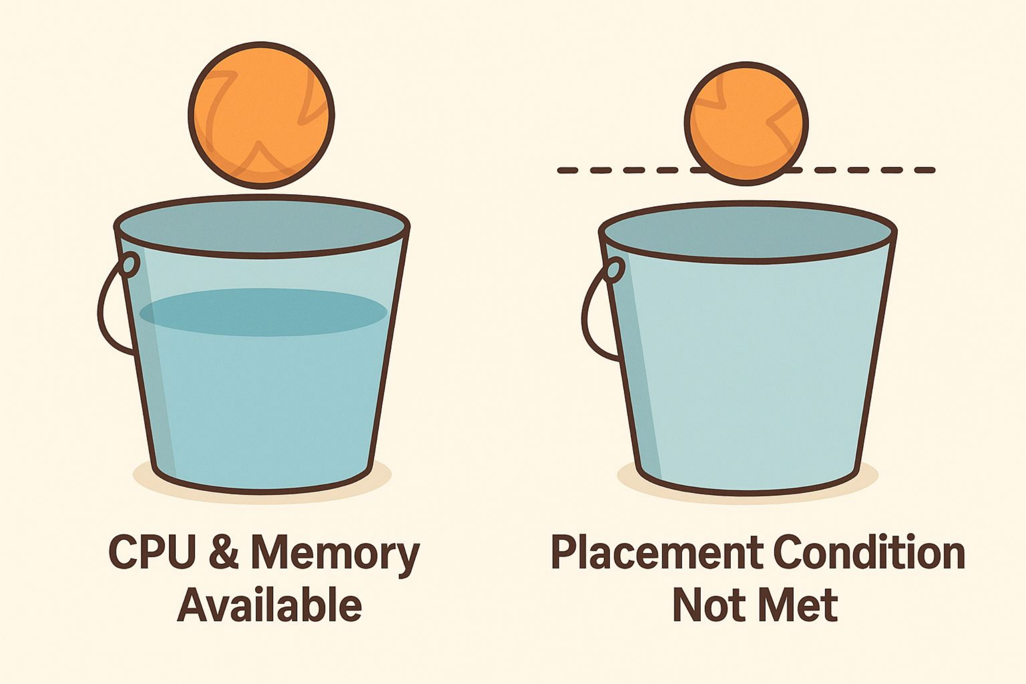
Today we'll solve the mystery of why tasks won't launch

We had 3 EC2s. Plenty of resources.

Still tasks wouldn't launch.

It Wasn't a Resource Issue.

It Was a Placement Problem



Agenda

→ The Hidden Reason your ECS tasks won’t launch

→ How the ECS Scheduler really thinks

→ Strategies vs. Constraints – and when to use them

→ Real-world failure & fix (our case study)

→ Checklist to prevent PENDING hell

→ Takeaways + Action plan

What is scaling ?

Vertical scaling

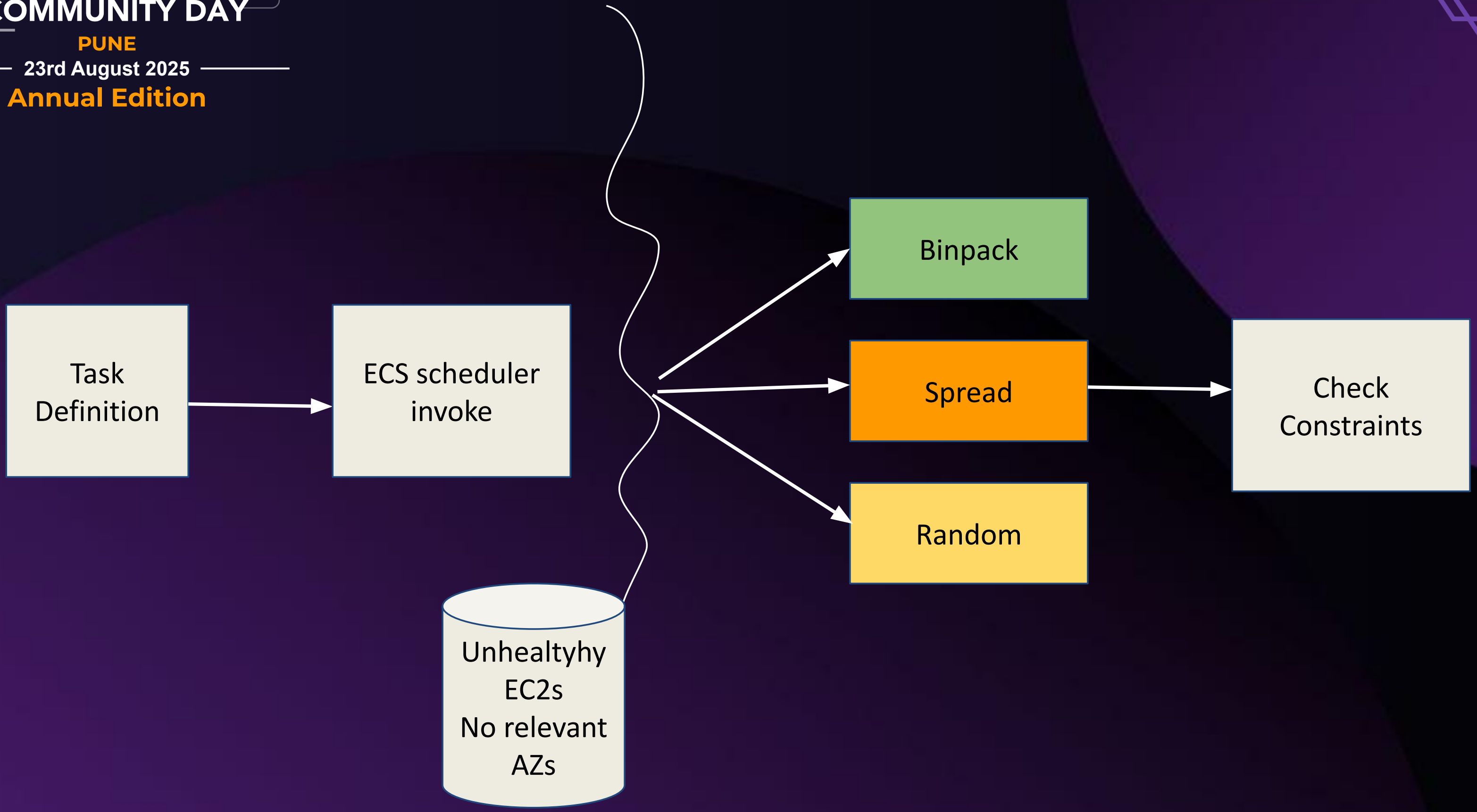
Horizontal scaling

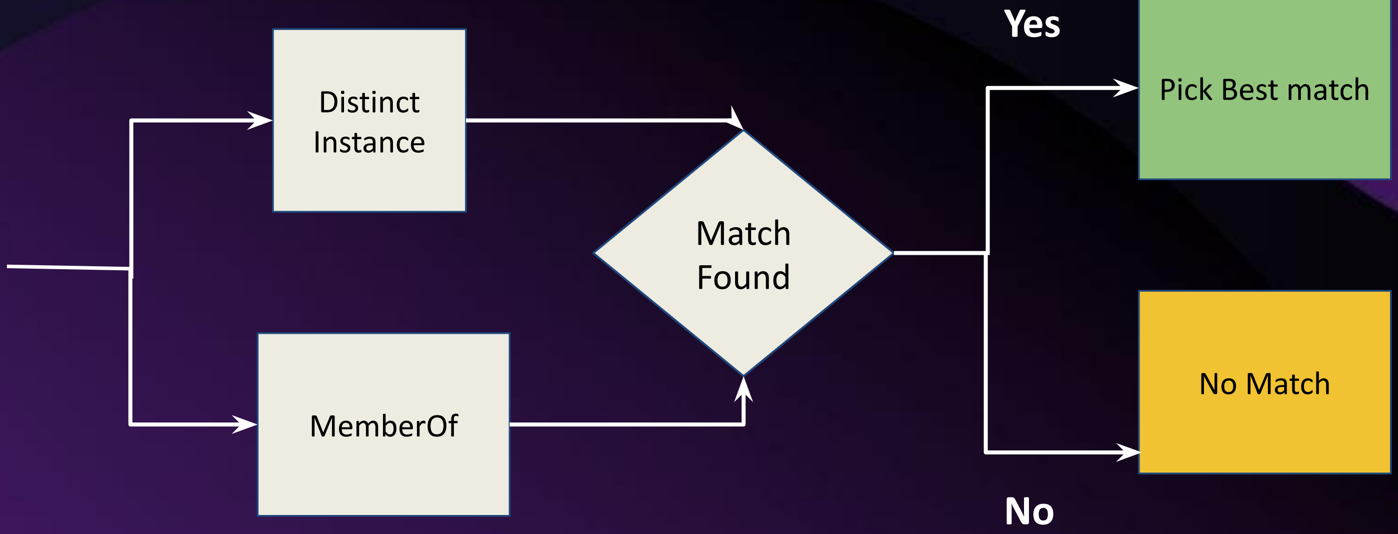
Bigger box more slices

More boxes more slices

Scaling isn’t just about adding CPU - it’s about placing tasks intelligently

How ECS places tasks?


Task Definition
ECS scheduler invoke
Unhealtyhy EC2s
No relevant AZs
Binpack
Spread
Random
Check Constraints


Distinct Instance
MemberOf
Match Found
Yes
Pick Best match
No
No Match

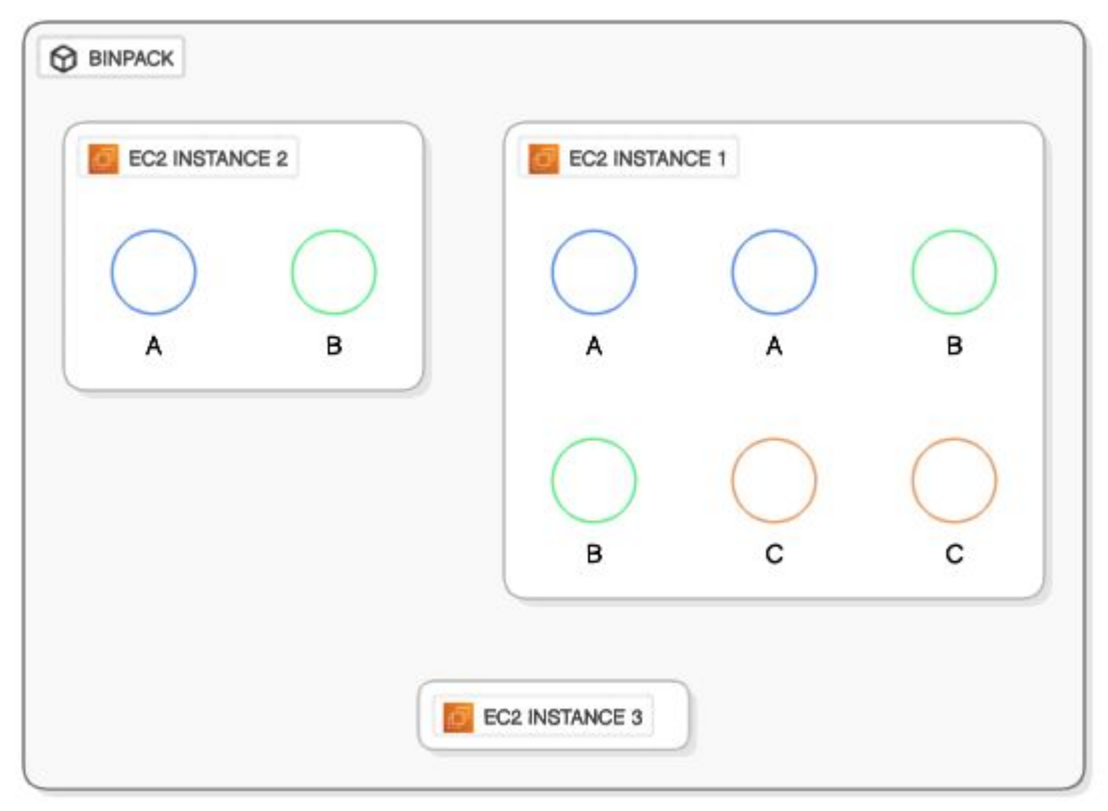
“Default isn’t always smart”

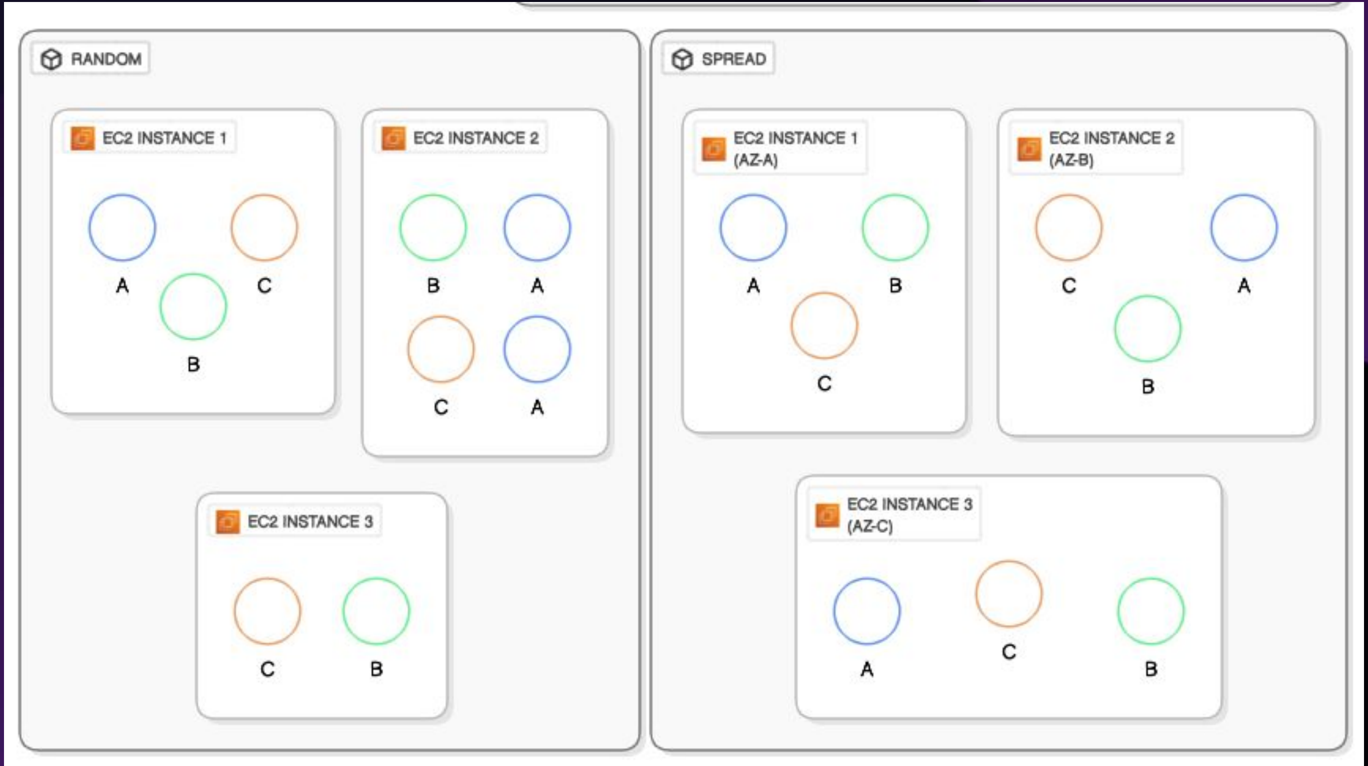
The default is a suggestion, not a strategy

Binpacked

Spread

Random


BINPACK
EC2 INSTANCE 2
A
B
EC2 INSTANCE 1
A
A
B
B
C
C
EC2 INSTANCE 3


RANDOM
EC2 INSTANCE 1
A
C
B
EC2 INSTANCE 2
B
A
C
A
EC2 INSTANCE 3
C
B
SPREAD
EC2 INSTANCE 1 (AZ-A)
A
B
C
EC2 INSTANCE 2 (AZ-B)
C
A
B
EC2 INSTANCE 3 (AZ-C)
A
C
B

When to Use Which Strategy?

Quick gut-check

- ➔ Binpack → cut costs, fewer hosts, better packing.
- ➔ Spread → uptime/resilience first, isolate risk across AZs/hosts.
- ➔ Random → good enough when you don’t care; baseline for noisy fleets.

The Trade-offs

- → 2 EC2s, 8 tasks
- → Cost: 💰 lower
- → Blast radius: 🔥 higher if instance fails

Binpack

- → 4 EC2s, 8 tasks
- → Cost: 💸 higher
- → Fault tolerance: ✅ higher

Spread

There’s no right answer.

It’s about your workload goals

When You Mix Strategy with Constraints ??

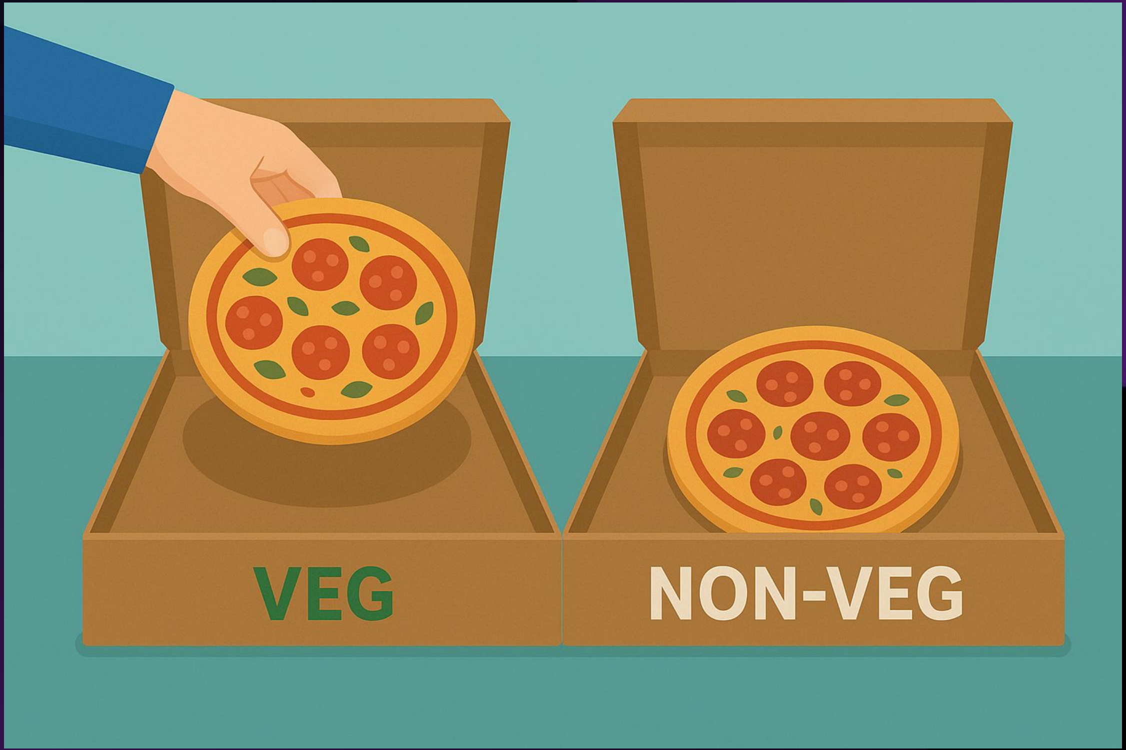
Placement Constraints

Distinct Instance

Why it matters

- ➔ If one apartment (host) burns down, you only lose one roommate (task).
- ➔ Great for high availability and avoiding “all eggs in one basket.”

MemberOf

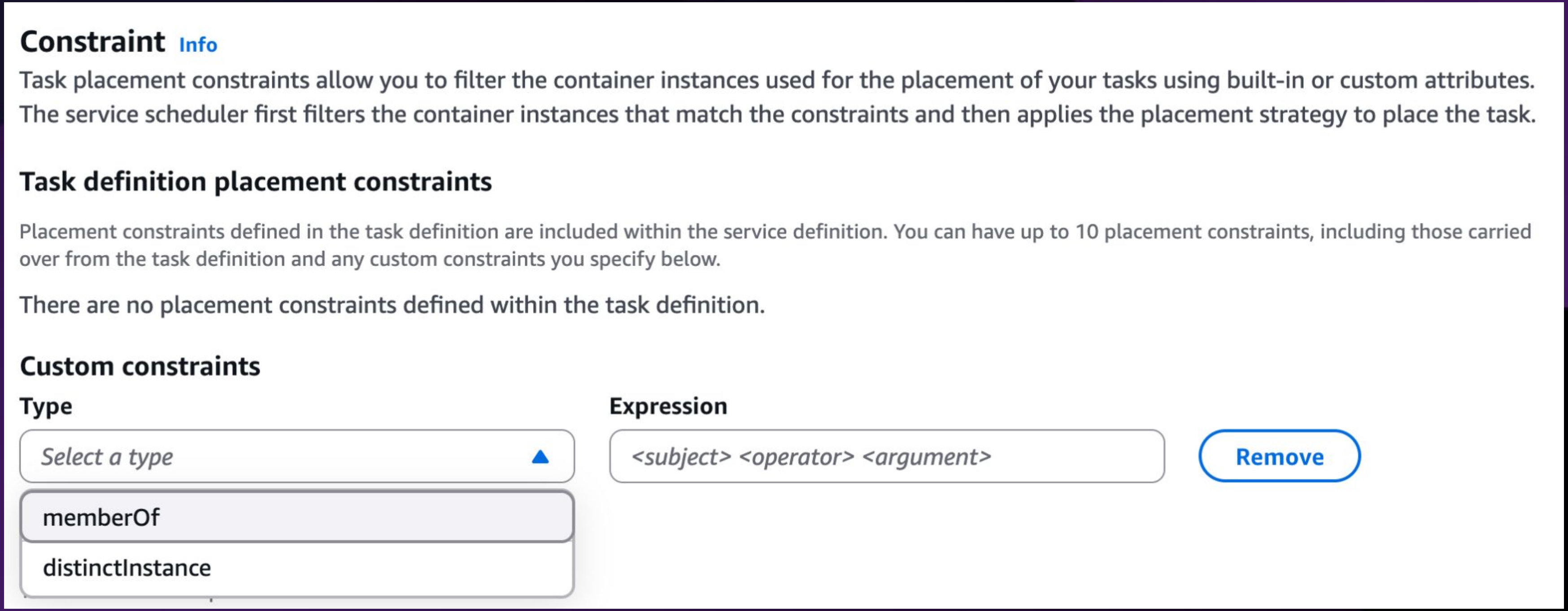


Why it matters

→ It stops AWS from placing your tasks anywhere.

→ Ensures tasks land only on the right kind of servers.

→ Without MemberOf, ECS might scatter your special tasks across the wrong instances. That wastes money and creates risks.



Constraint Info

Task placement constraints allow you to filter the container instances used for the placement of your tasks using built-in or custom attributes. The service scheduler first filters the container instances that match the constraints and then applies the placement strategy to place the task.

Task definition placement constraints

Placement constraints defined in the task definition are included within the service definition. You can have up to 10 placement constraints, including those carried over from the task definition and any custom constraints you specify below.

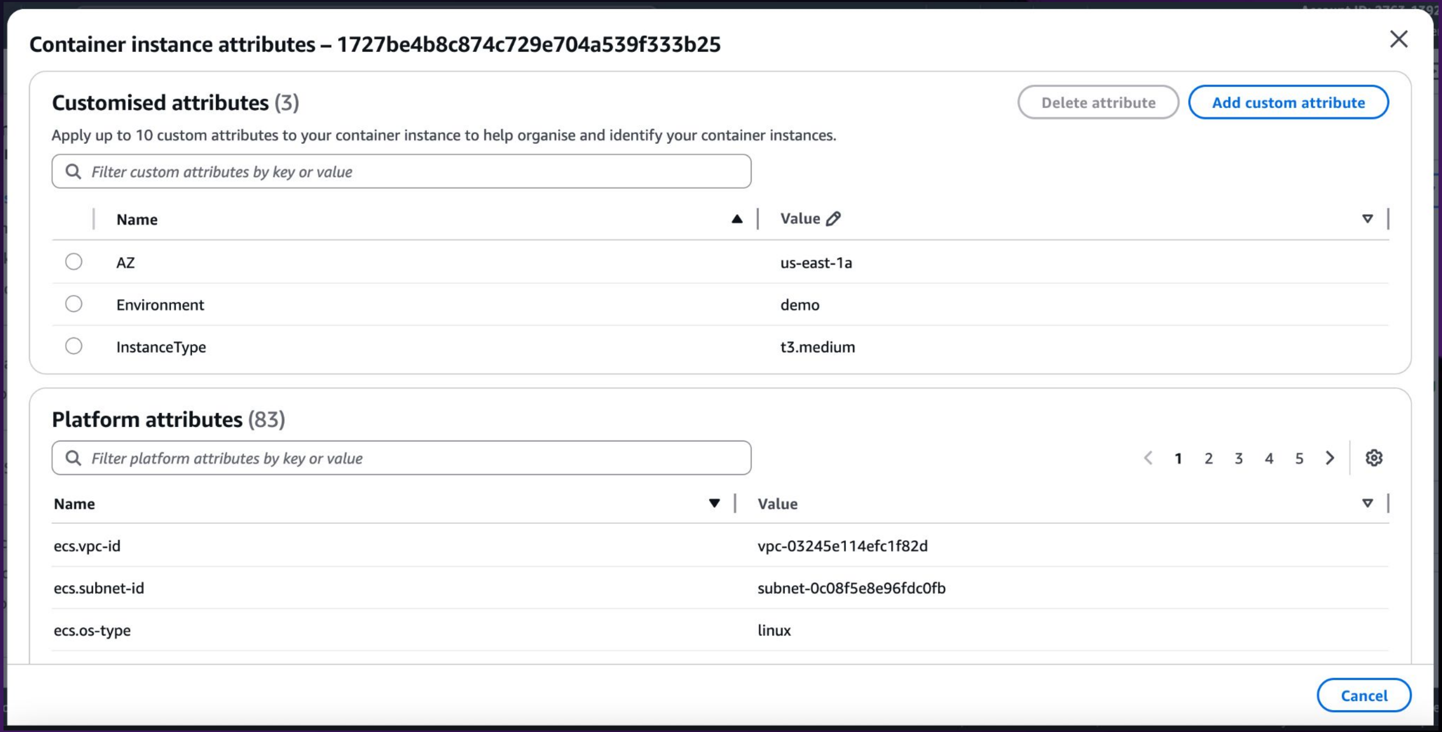
There are no placement constraints defined within the task definition.

Custom constraints

Type: *Select a type* ▲

- memberOf
- distinctInstance

Expression: *<subject> <operator> <argument>*



Container instance attributes – 1727be4b8c874c729e704a539f333b25

Customised attributes (3)

Delete attribute
Add custom attribute

Apply up to 10 custom attributes to your container instance to help organise and identify your container instances.

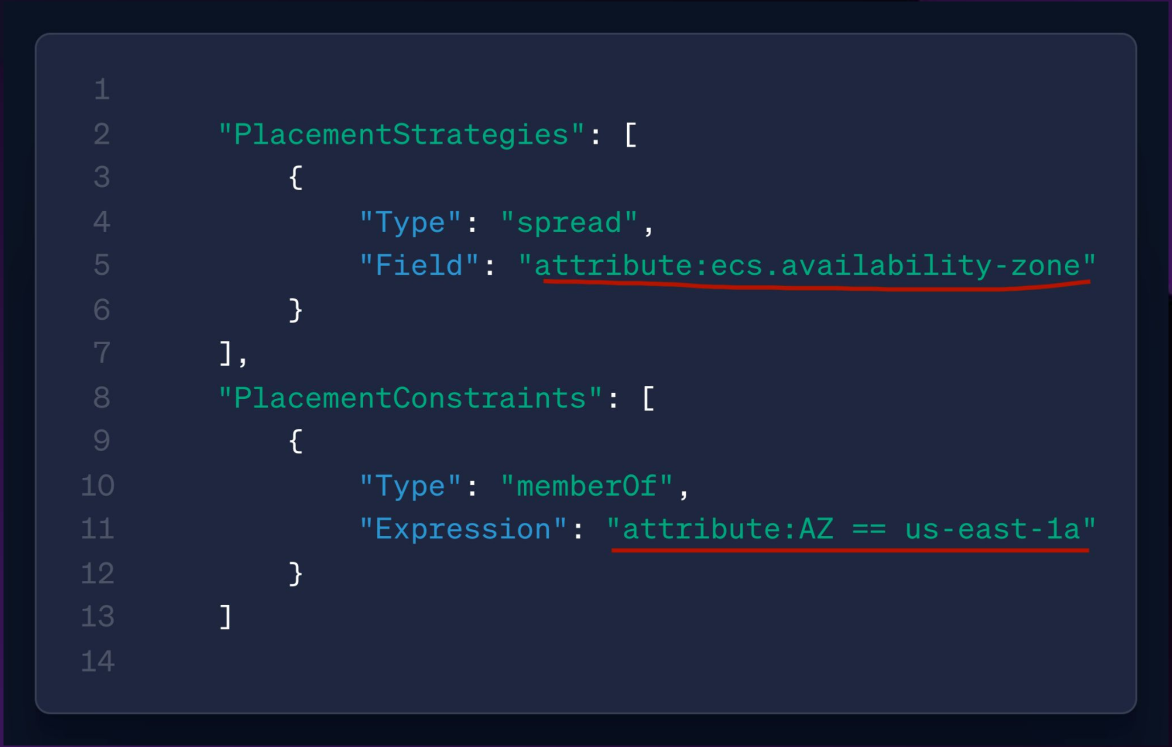
Filter custom attributes by key or value

	Name	Value
○	AZ	us-east-1a
○	Environment	demo
○	InstanceType	t3.medium

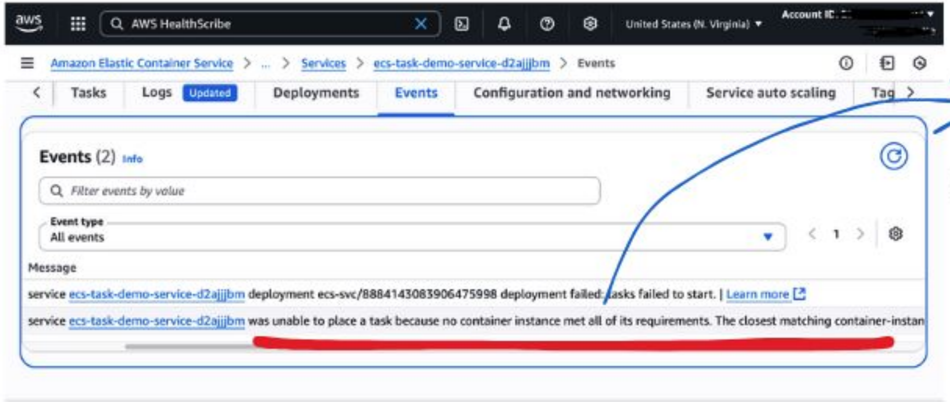
Platform attributes (83)

Filter platform attributes by key or value
< 1 2 3 4 5 >

Name	Value
ecs.vpc-id	vpc-03245e114efc1f82d
ecs.subnet-id	subnet-0c08f5e8e96fdc0fb
ecs.os-type	linux



```
    "PlacementStrategies": [
        {
            "Type": "spread",
            "Field": "attribute:ecs.availability-zone"
        }
    ],
    "PlacementConstraints": [
        {
            "Type": "memberOf",
            "Expression": "attribute:AZ == us-east-1a"
        }
    ]
```



Message

service ecs-task-demo-service-d2ajjjbm deployment ecs-svc/8884143083906475998 deployment failed: tasks failed to start. | Learn more

service ecs-task-demo-service-d2ajjjbm was unable to place a task because no container instance met all of its requirements. The closest matching container-instan

It says no container instance met all requirements. The closest one was missing an attribute. That was the exact hint we needed.

Gotchas that cause PENDING State

- ❌ Placement strategy + constraint mismatch
- ❌ Task resource requests > available instance capacity
- ❌ EC2 instance health issues
- ❌ AZ imbalance or missing capacity
- ❌ Constraint on nonexistent attribute (e.g., forgot to tag instance)

What I Want You to Take Away

→ ECS doesn’t just place tasks — it makes decisions based on your strategy

→ Cost vs resilience is a tradeoff you get to design

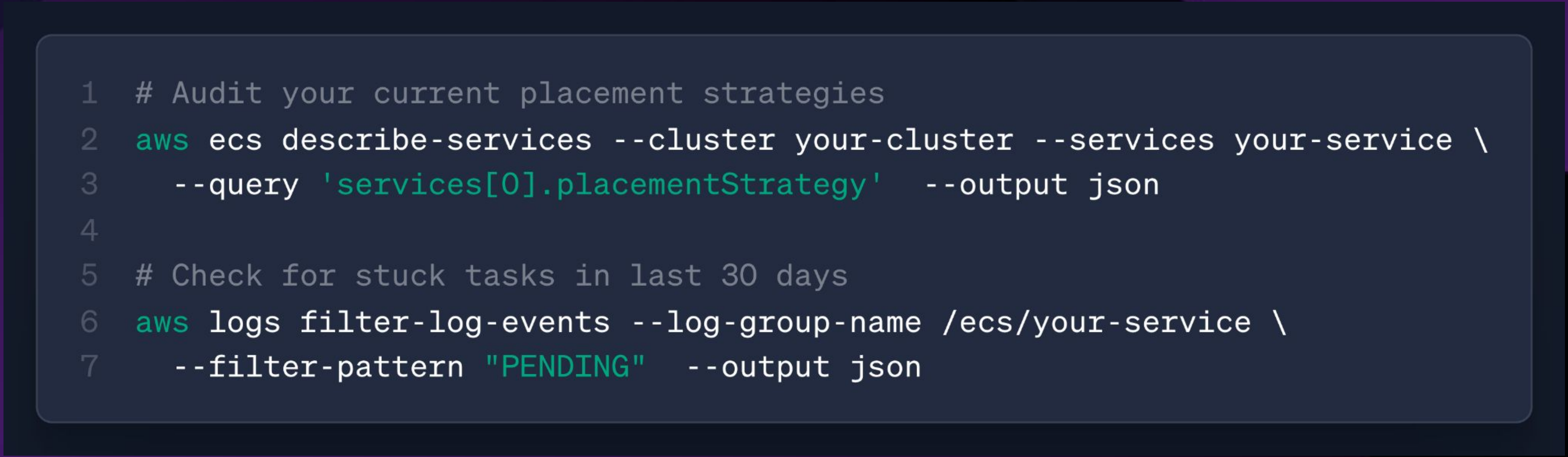
→ If tasks are stuck, don’t guess — decode the scheduler

→ Your strategy + constraints = your architecture

What Will You Do Differently?

- ➔ Try switching from spread to binpack or vice versa — and measure the result
- ➔ Check your ECS Events tab next time something is “**just stuck**”
- ➔ Share your story. ECS is simple, but not easy.

Your Action Plan - Starting Tomorrow



```
# Audit your current placement strategies
aws ecs describe-services --cluster your-cluster --services your-service \
  --query 'services[0].placementStrategy'  --output json

# Check for stuck tasks in last 30 days
aws logs filter-log-events --log-group-name /ecs/your-service \
  --filter-pattern "PENDING"  --output json
```

Common Placement Failure Scenarios

➔ Insufficient Resources → No host meets CPU/mem needs

➔ Constraint Mismatch → Strategy vs. attribute conflict

➔ Port Conflicts → Static port already in use

➔ more cases details are available here https://github.com/AvinashDalvi89/ecs-task-placement-guide

ECS gives you the power to place workloads intelligently.

The question isn't whether you can scale.

It's whether you can scale **smartly**.

Q&A ??

aws COMMUNITY DAY

PUNE

23rd August 2025

Annual Edition

aws USER GROUP PUNE

aws

COMMUNITY DAY

PUNE

23rd August 2025

Annual Edition

aws

USER GROUP

PUNE

Thank you!