Turbocharge ECS Fargate

SOCI Indexed Containers

1. **Introduction**
    a. SOCI?

2. **SOCI preview**
    a. Testing
    b. Issues

3. **SOCI GA**
    a. Testing
    b. Issues
    c. AWS meeting

4. **Results**
    a. Task start times
    b. Bulk task start times

5. **Implementation**
    a. Memory requirements
    b. CF Template

6. **Next steps**
    a. Retune service scaling

SECTION 1

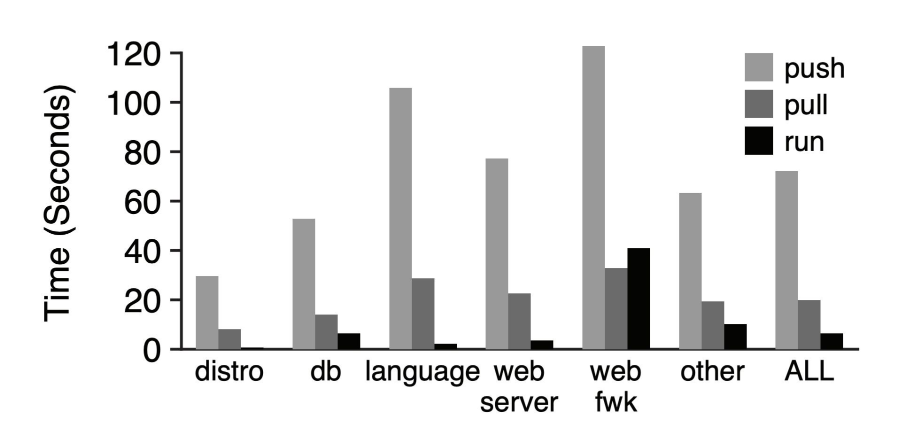
What is SOCI?

What is SOCI?

What is SOCI?

it’s pronounced so-chee

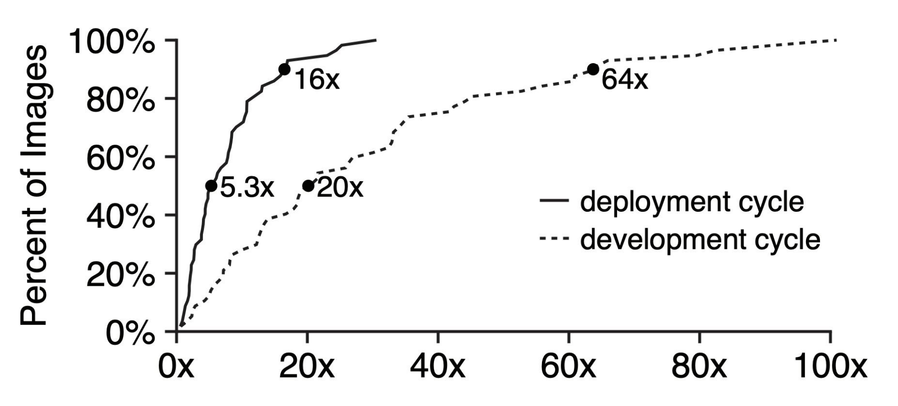
- Seekable OCI images
- 76% of container start-up time is pulling an image
- 6.4% of container data is actually needed to start running



Celtra

Lazy loading of container images

- Create an index and pull data only when needed
- Speed up container startup time by 5x



Celtra

SECTION 2

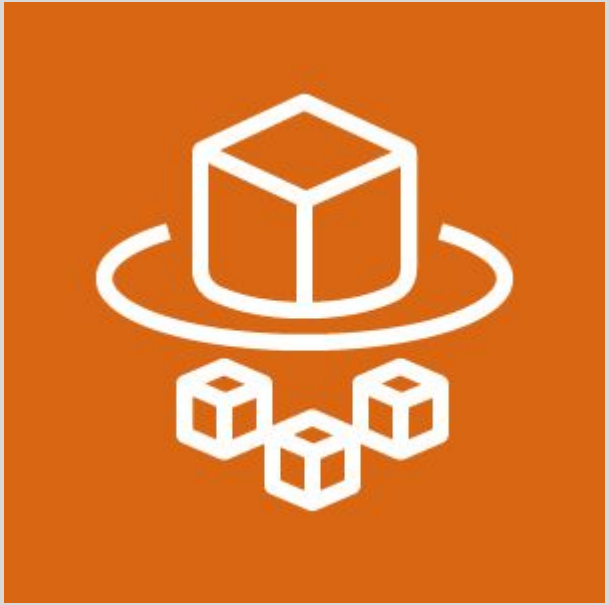
SOCI preview

AWS invites us to a private preview

- March 2023
- ECS on Fargate
- Use containerd to create index
- Push to ECR and tasks use index automatically
- soci index is tagged by digest of the image

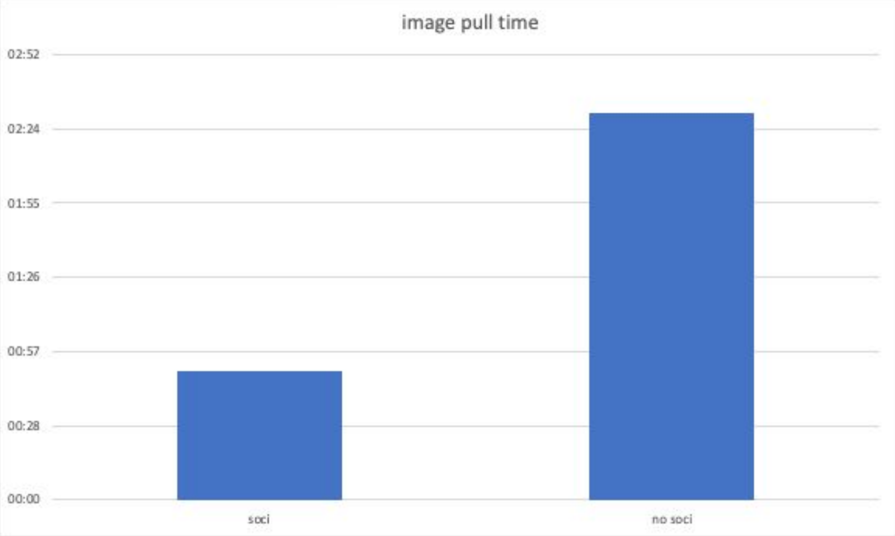
SOCI preview issues

- ECS Fargate
- No ARM support
- No firelens support



SOCI preview testing

- Container image running on Fargate
- No use case unless ARM and Firelens are supported
- Looks promising



Celtra

SECTION 3

SOCI GA

SOCI GA Reevaluation

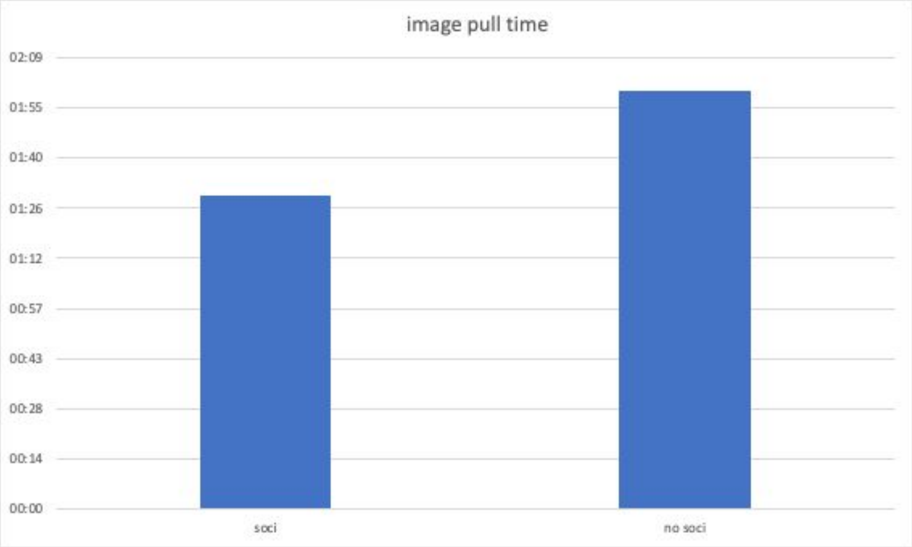
- July 2023
- ARM and Firelens images are supported
- No apparent blockers for now
- All container images in a task must have index
- Easier setup

Testing SOCI on fullstack

- Less speed up
- Still 25% speed up

- Some services do not start up
- Stuck in pending forever

Celtra



AWS Support

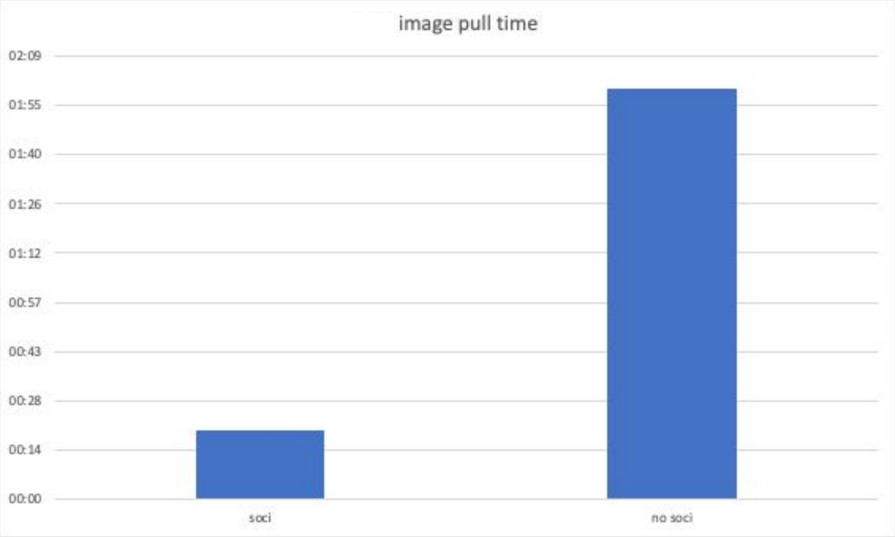
- Not enough resources allocated
- A call is scheduled
- Key takeaways
  - Index is not required for all images
  - Update SOCI snapshotter
  - Additional memory usage only in pull time

SECTION 3

Reevaluation

SOCI fullstack retest

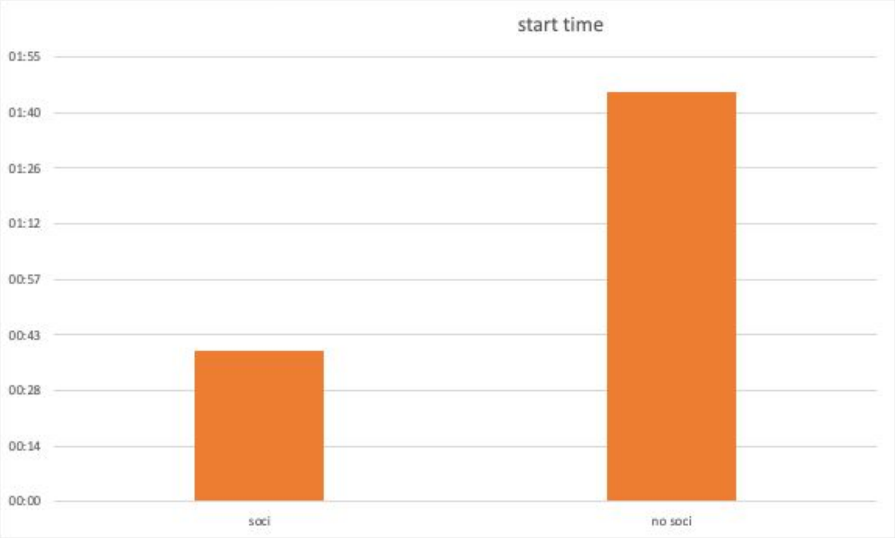
- 1GB of memory per task
- Pull time reduced to 20s
- Task start time under one minute



Celtra

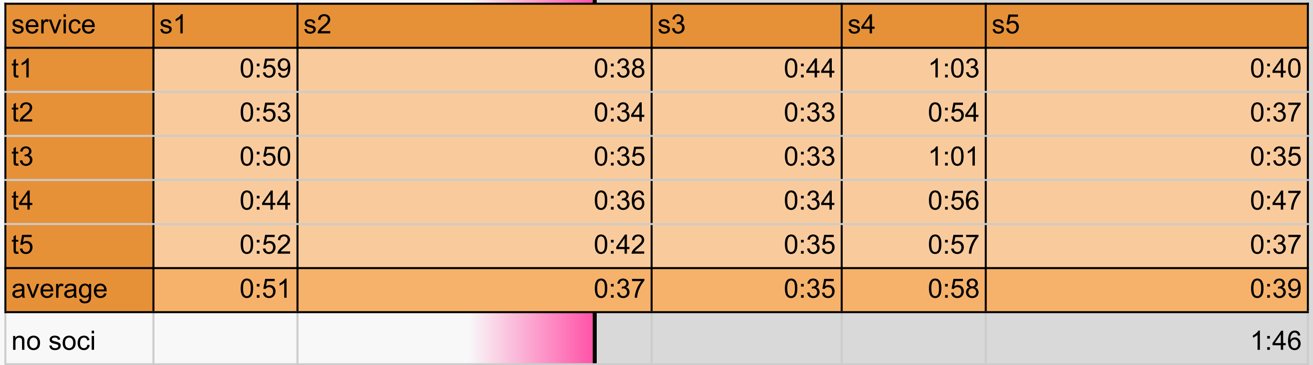
SOCI UAB fullstack scaling

- 1GB of memory per task
- Pull time reduced to 20s
- Task start time under one minute



Celtra

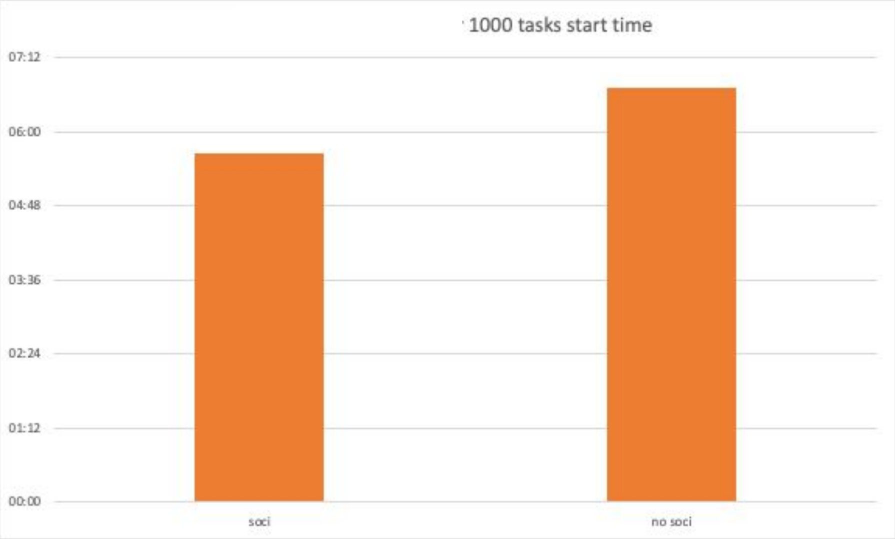
SOCI fullstack scaling



service	s1	s2	s3	s4	s5
t1	0:59	0:38	0:44	1:03	0:40
t2	0:53	0:34	0:33	0:54	0:37
t3	0:50	0:35	0:33	1:01	0:35
t4	0:44	0:36	0:34	0:56	0:47
t5	0:52	0:42	0:35	0:57	0:37
average	0:51	0:37	0:35	0:58	0:39
no soci					1:46

SOCI UAB fullstack bulk scaling

- Bulk scaling sees less improvement
- Not enough pre-warmed ARM Fargate instances

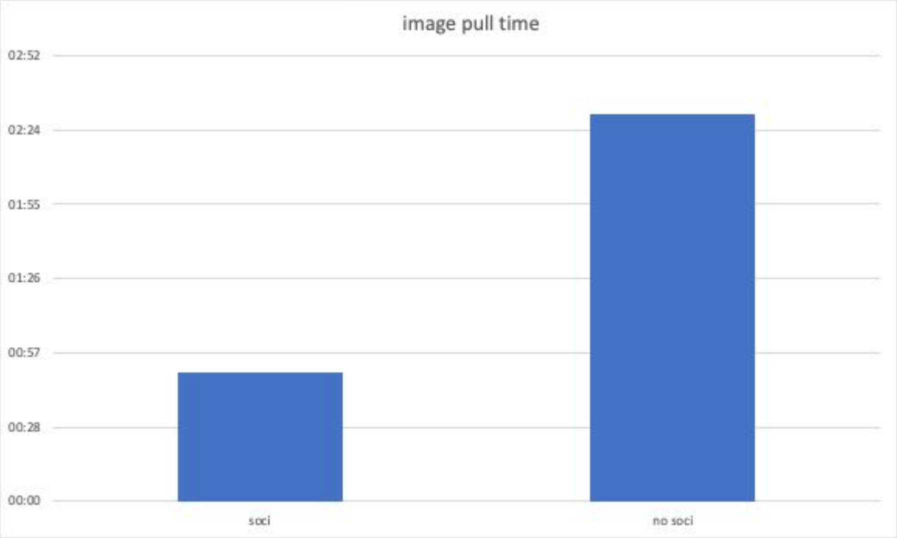


Celtra

SOCI with a larger image

- SOCI lazy loading available only for ECS on Fargate

- 1:20 for php service
- 1:30 for scala service
- 1:35 for 100 scala services



Celtra

SECTION 5

Implementation

Implementing SOCI to production

- Cloudformation template provided by AWS
  - Event bridge rule
  - Filtering lambda
  - SOCI index generator lambda
- 2GB of memory per service

SECTION 6

Next steps

Super fast scaling?

- Our bottleneck for scaling is now AWS
- 3 consecutive breaches for autoscaling to kick in
  - 3 minutes
  - rethink how we scale services
- not enough ARM instances

Celtra

References

- https://www.usenix.org/system/files/conference/fast16/fast16-papers-harter.pdf
- https://aws.amazon.com/blogs/aws/aws-fargate-enables-faster-container-startup-using-seekable-oci/

Thank you!

Celtra

Celtra