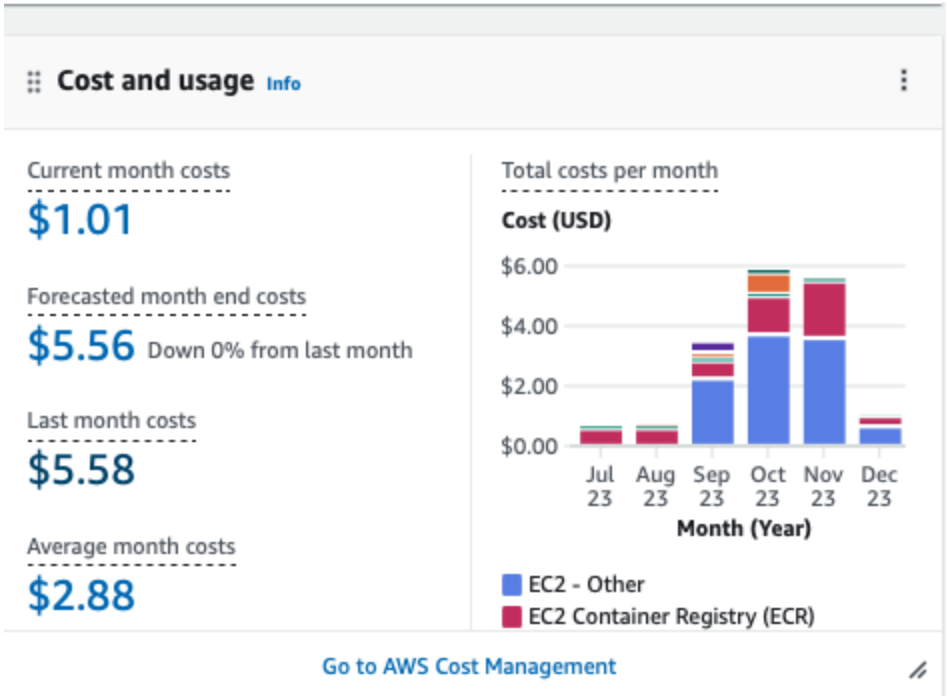
RoboMaker Discussion

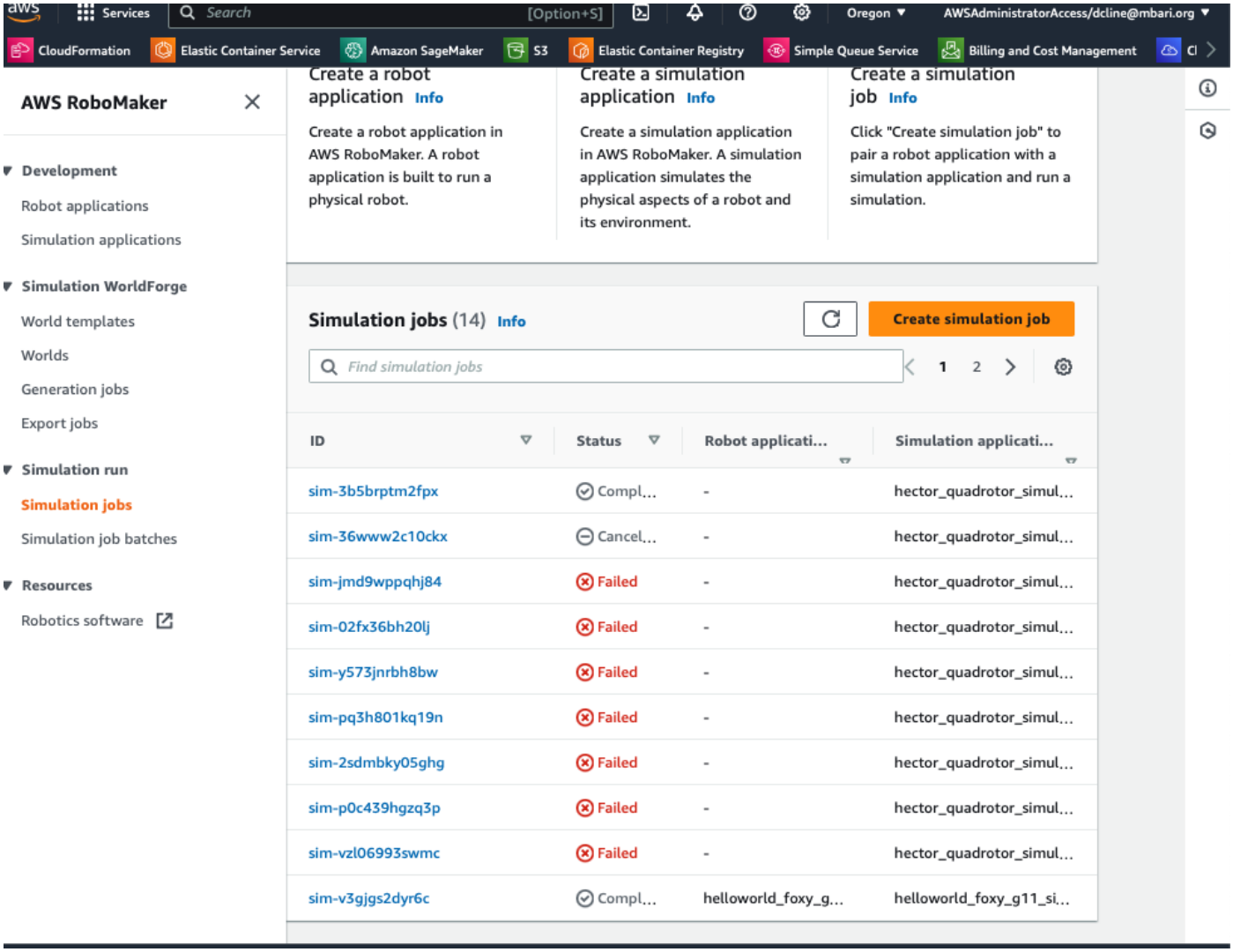
12-06-23

**1. Introduction

- Brief overview of AWS RoboMaker
  - "AWS RoboMaker is a cloud-based simulation service that enables robotics developers to run, scale, and automate simulation without managing any infrastructure." - AWS
  - This hides the complexities of spinning up and down compute/storage infrastructure, logging all simulation output, etc.
  - Pay as you go: Cost per SU **simulation unit**;
    - Minimum of 1vCPU and 2 GB memory = $0.40 per hour. Typically, a simulation may need 7 SU
    - GPUs are available too at $1.50 per hour
    - Costs for experimenting peaked at $6 in October



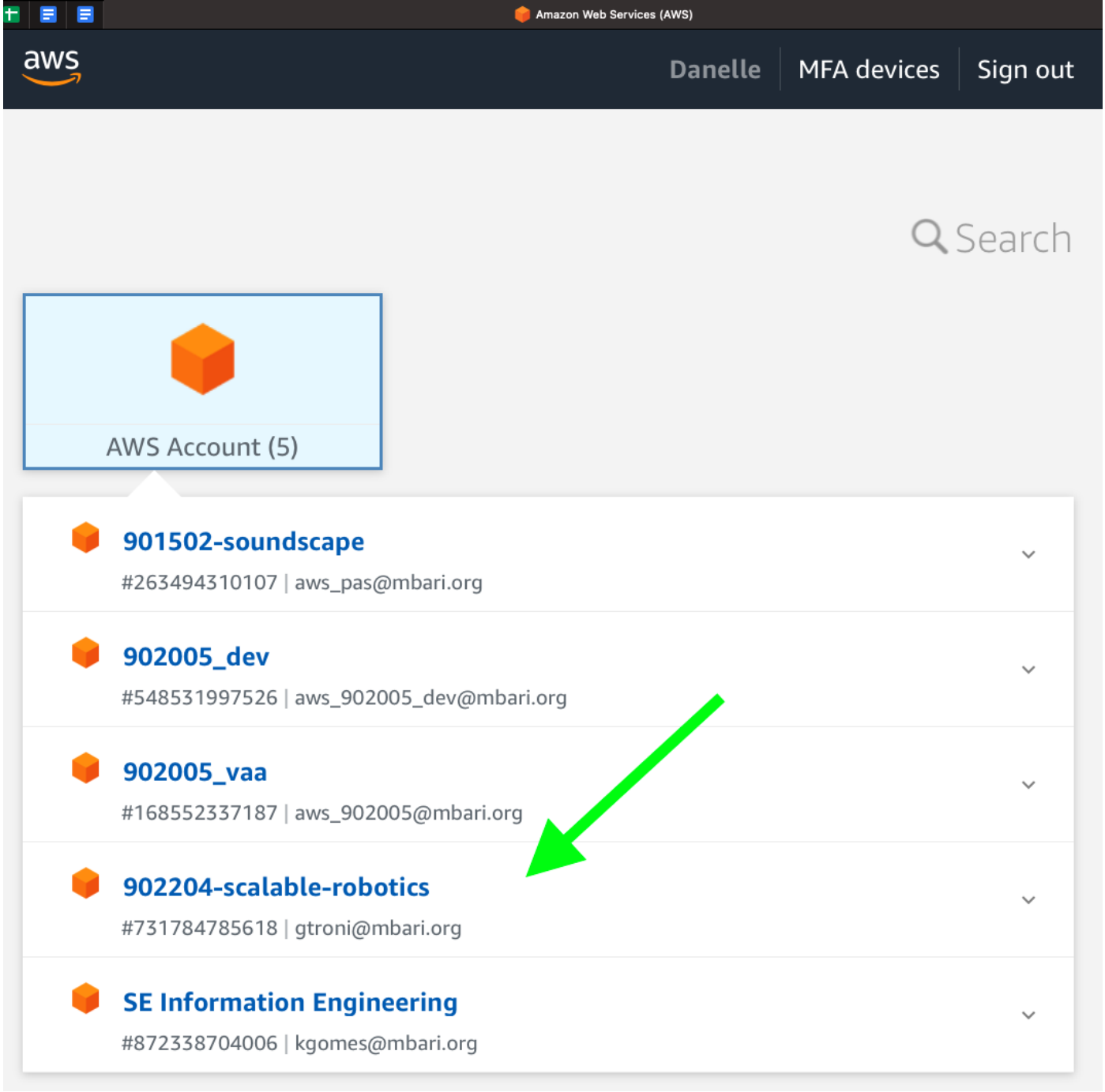
- Dashboard view



- Why is this important? What are the current limitations of bigtuna machine in the lab?
- **Agenda - walk-through of helloworld app followed by Q&A.**

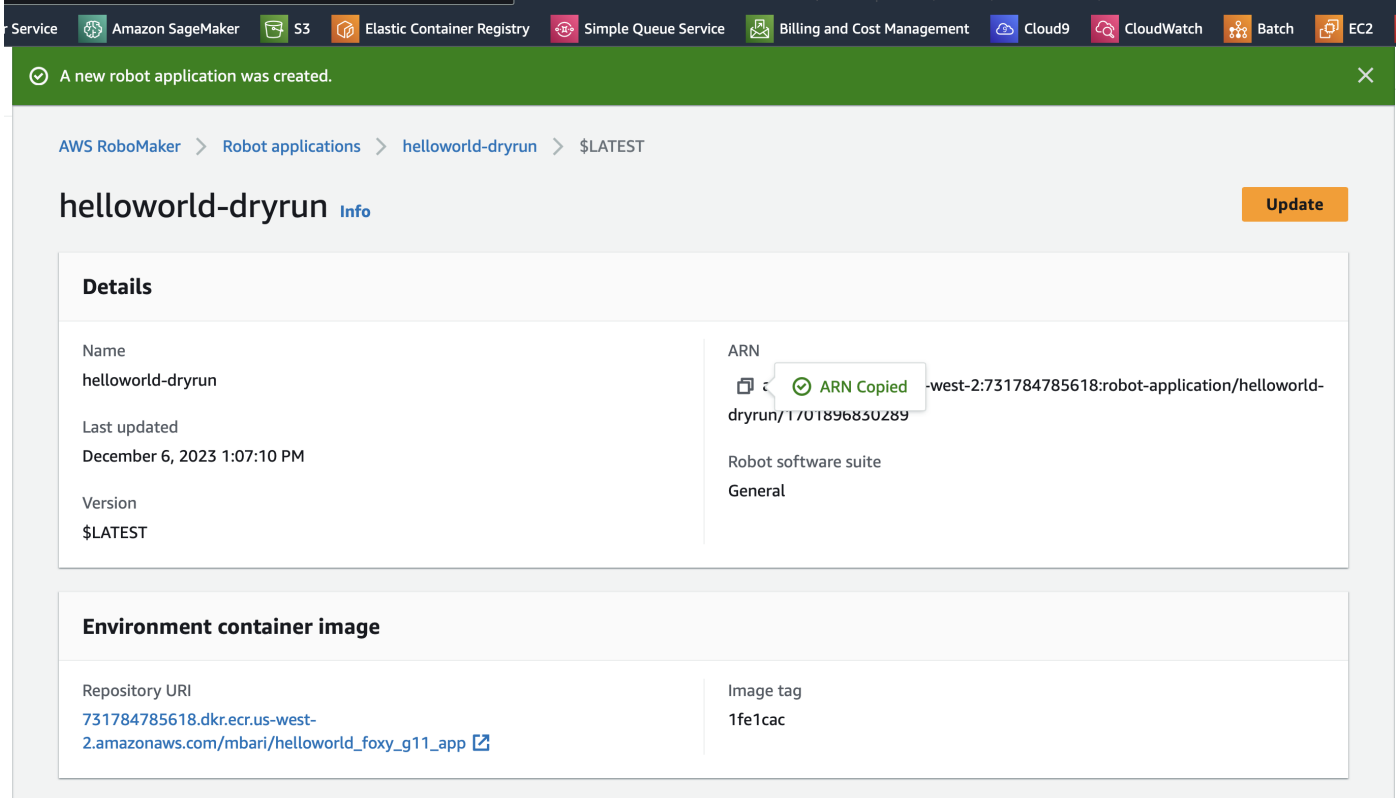
**2. Setting Up Your Environment

- Overview of required AWS services (RoboMaker, S3, IAM)
  - RoboMaker - managed service
  - S3 - Simple storage service (to log your results)
  - IAM - Identity Access management (to control who can do what)
- MBARI AWS account setup
  - IS setup and MBARI SSO - single sign-on at https://mbari.awsapps.com/start use your MBARI login ***Note that we can add** local* *users to this account to extend to collaborators*
  - Select **902204-scalable-robotics**, then select Management Console. This has a unique sub-account number **731784785618**, but still in the MBARI account.



**3. Creating Your First Robot Application

- Use RoboMaker IDE to create a simple **helloworld** robot application
  - Assume docker images are created and ready for use in the ECR (**Elastic Container Registry**). See the build_and_push_aws.sh in the robomaker-helloworld repo.
  - Can add tags to track costs
  - Each application creates has a unique ARN, e.g., a**rn:aws:robomaker:us-west-2:731784785618:robot-application/helloworld-dryrun/1701896830289** . *This is used when creating the simulation*



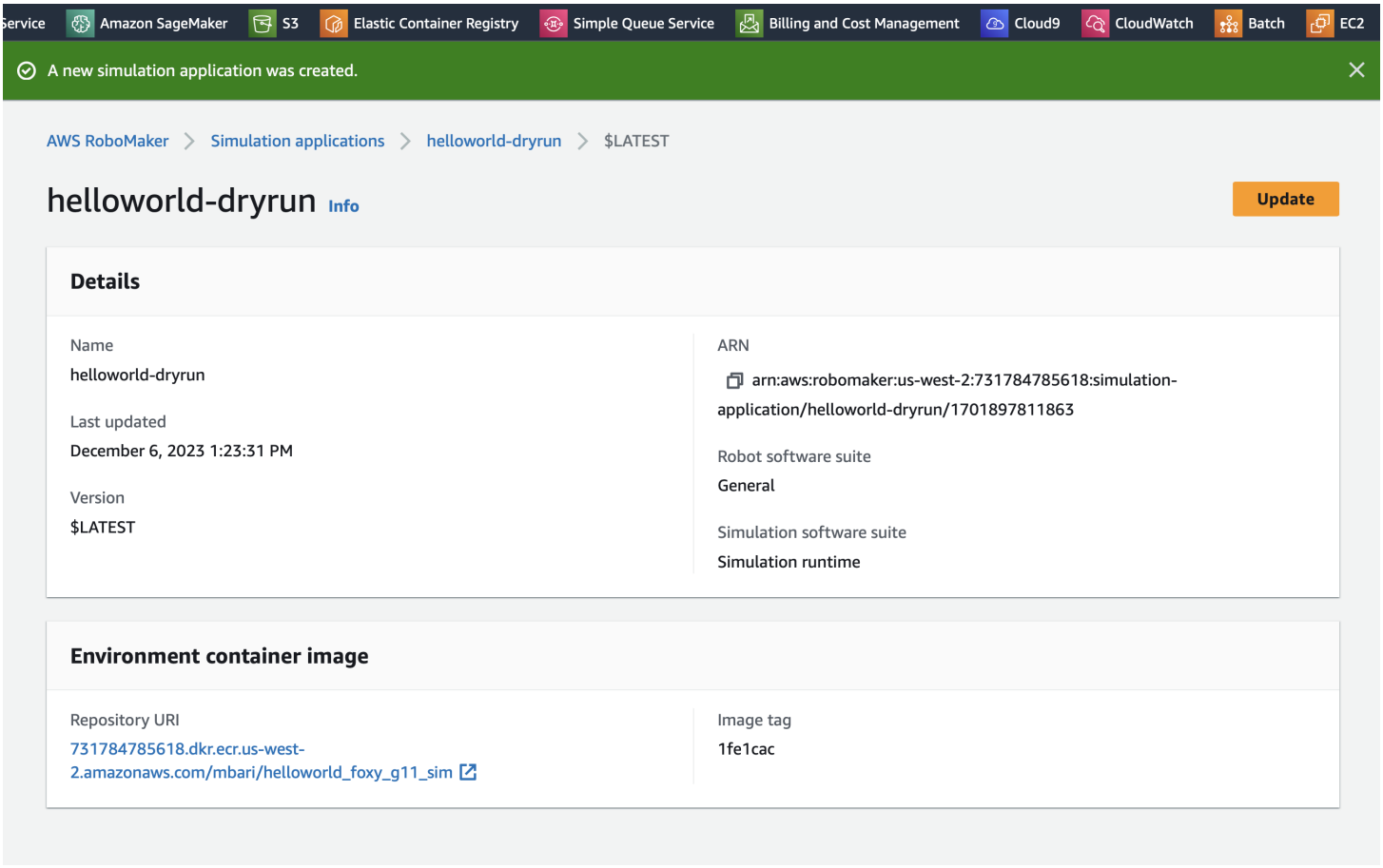
DISCUSSION

- Adding new features to the robot application
- Understanding robot application dependencies

4. Configuring Simulation

- Use RoboMaker IDE to create a simple **helloworld** robot simulation. Similarly to the application, there is a unique **ARN**. This is used in the **Simulation Job** (more on that later)

- Default storage is in a 128 GB partition



DISCUSSION

- **Logging**
  - Can adjust what is logged and what S3 bucket to store in the **Simulation Job**
  - Can query and tail stdout/err that is logged to the AWS Cloudwatch service
- **IP address assignment**
  - Can assign a public IP during simulation (just like the elastic inference engines).
- Configuring simulation settings in RoboMaker
- Adjusting simulation parameters

**5. Launching Simulation

- Deploying the robot application to the simulation environment as a **Simulation Job**
- Monitoring simulation progress in Cloud Watch

- D.Cline setup an IAM role called **robomaker-jobs** for this with appropriate permissions

**6. Q&A Session

**7. Conclusion and Next Steps

- Recap of key takeaways
  - RoboMaker can accelerate simulation
  - There is some overhead to putting your containers into a RoboMaker-compliant format
  - These docker images can be run locally
  - There is some overhead to managing the output, e.g., in S3 and viewing the logs

- What do we want to do ?
  - Shelf the exploration
  - Scale caleucheenv_rl simulations
  - Others