
AWS
re:Invent

ROB202

Build a better robot faster with AWS and ROS

Matthew Hansen

Principal Solutions Architect, Robotics

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Intro to AWS RoboMaker using ROS (Robot Operating System)

ROS overview and benefits

ROS concepts and example

Gazebo simulation overview

Robot software development cycle

AWS RoboMaker overview and demo

Conclusion, links, and contact info



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Building robots is HARD!



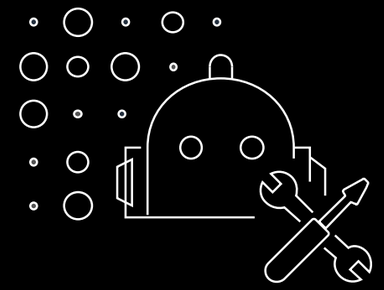
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introducing AWS RoboMaker

A service that makes it easy for developers to develop, test, and deploy robotics applications, as well as build intelligent robotics functions using cloud services



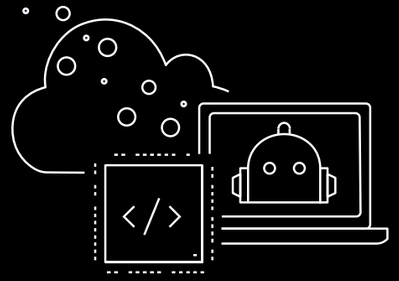
Intro to AWS RoboMaker using ROS



ROS open-source tools and cloud extensions



Simulation



Development environment



Fleet management

ROS overview

What is 

?

www.ros.org

ROS is an open-source framework that provides inter-process communication mechanisms, standard message types, robotic control and observation tools, hardware abstractions, and common algorithms for robot software development.

ROS benefits

Why ROS?

Distributed architecture – loose coupling

Standard messages – hardware abstraction

Established community – over 3,000 community-contributed packages

Simulation interfaces – decouple hardware from testing algorithms

Building Robots is HARD!

ROS makes it easy to reuse and extend existing software so that developers can focus on new algorithms, hardware, and use cases.

ROS key concepts

Nodes
Processes that perform computations
(Ex: your Python files)

ROS Core
Provides name registration and lookup to the rest of the Computation Graph. *Without the Core, nodes would not be able to find each other, exchange messages, or invoke services*

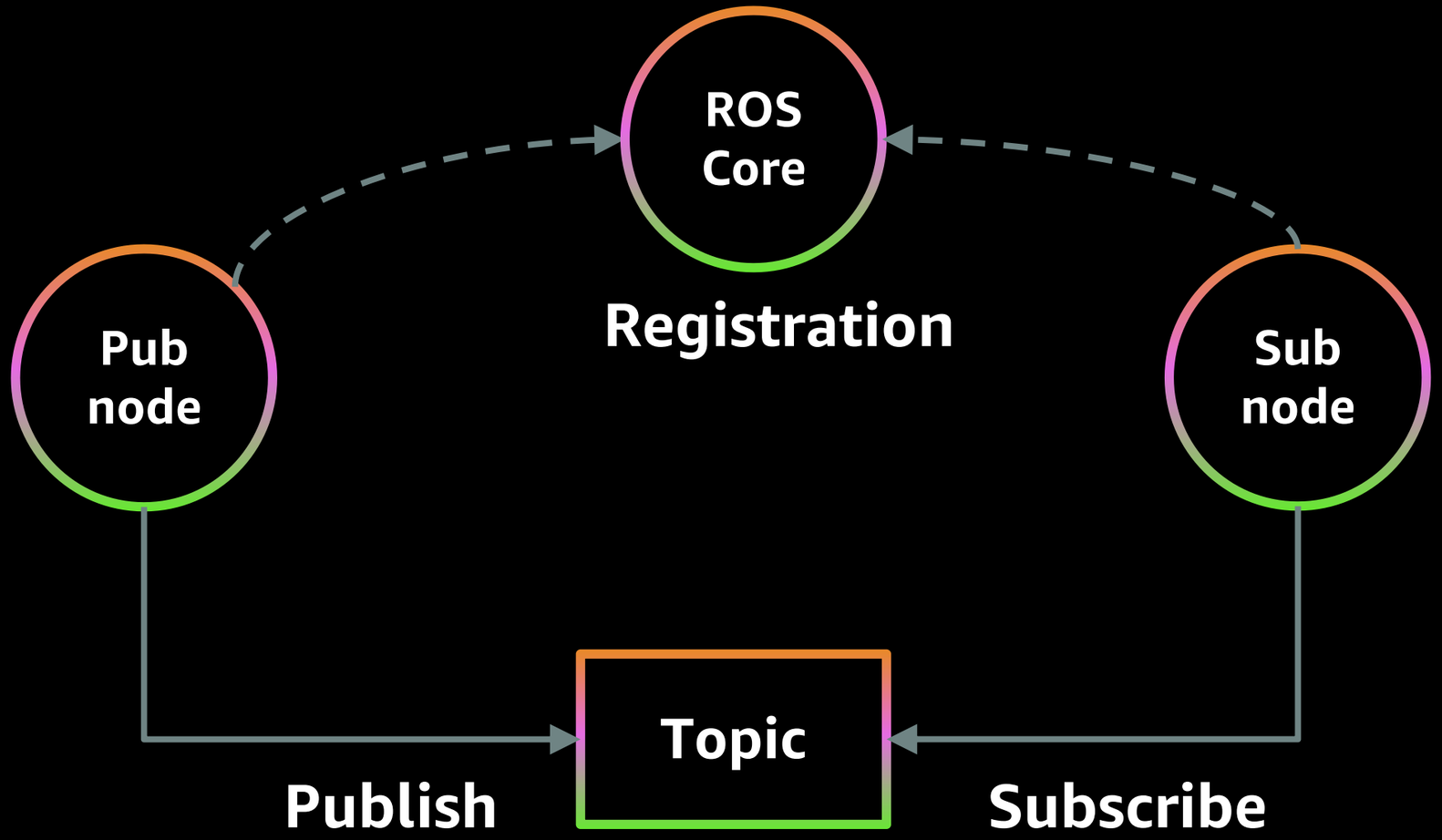
Messages
A message is simply a data structure, comprising typed fields

Topics
Messages are routed via a transport system with publish / subscribe semantics. A node sends out a message by publishing it to a topic

Services
Request / reply is done via services, which are defined by a pair of message structures: one for the request and one for the reply

ROS Bags
Bags are a format for saving and playing back ROS message data

ROS publish / subscribe example



ROS remote procedure calls

There are two ways nodes can call other nodes to perform a computation:

Services

- Synchronous / blocking functions
- Useful for getting / setting state information in another node

Example: Check if hardware is enabled

Actions

- Asynchronous / non-blocking functions
- Useful for long-running processes

Example: Send a goal location to a robot and get a response later when completed



Common ROS topics

/move_base_simple/goal (geometry_msgs/PoseStamped)

Send a goal pose to robot for autonomous navigation

/cmd_vel (geometry_msgs/Twist)

Command robot to move with given directional and rotational velocities

/map (nav_msgs/OccupancyGrid)

Provides the current occupancy grid, or map representation

/odom (nav_msgs/Odometry)

Odometry is change in position over time compared to starting location

/scan (sensor_msgs/LaserScan)

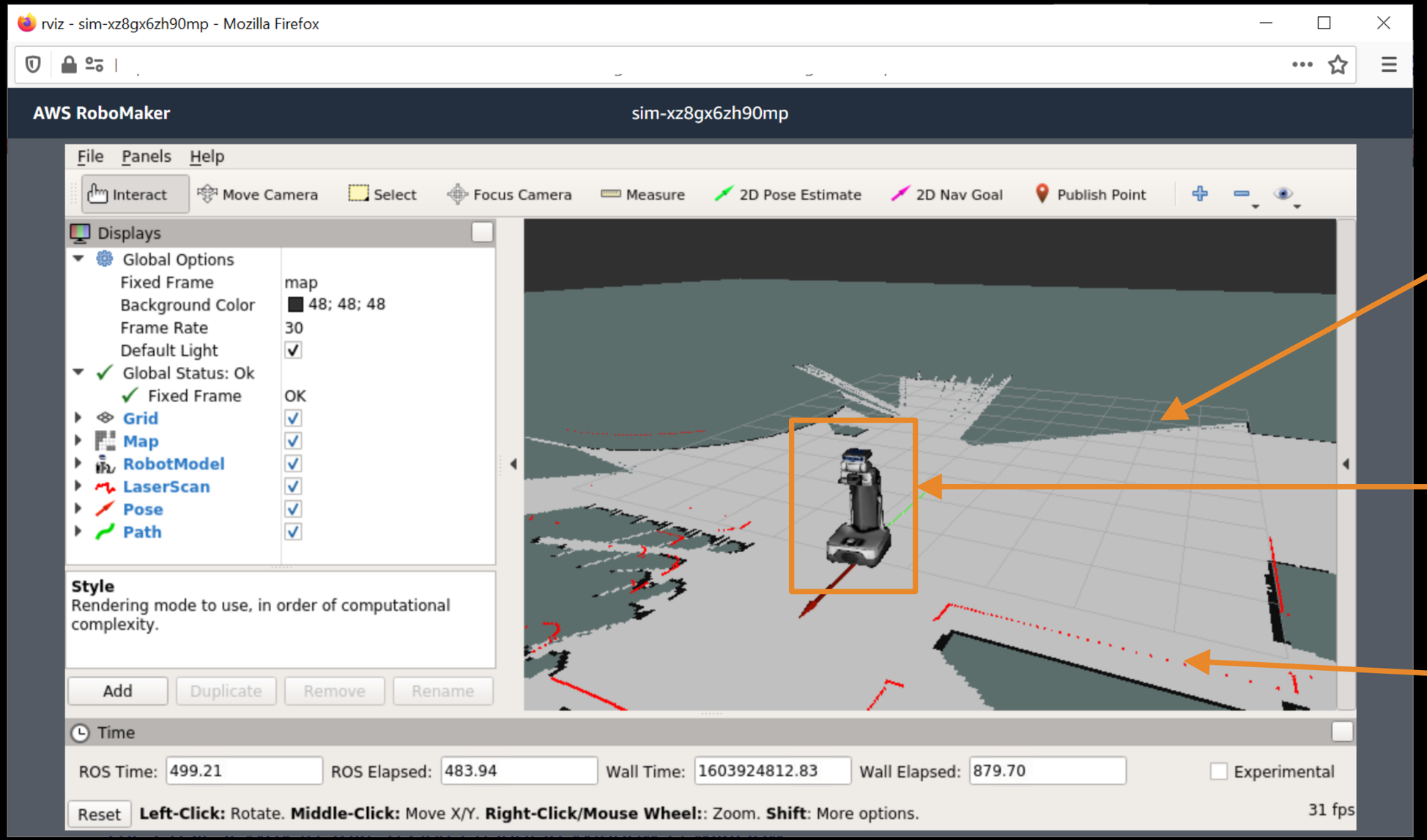
Laser scan data from sensors like LiDARs

/camera/image_raw (sensor_msgs/Image)

Image frames coming from a camera feed



ROS visualization – RViz



Map

Robot model

Laser scan

ROS launch

- Spawns multiple ROS nodes running in parallel
- Sets parameters in the parameter server
- XML (ROS1), Python (ROS2)
- AWS RoboMaker simulation uses ROS launch to start your applications

```
<launch>
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name"
           value="$(find aws_robomaker_hospital_world)/worlds/hospital.world"/>
    <arg name="use_sim_time" value="true"/>
  </include>
  <include file="$(find pr2_gazebo)/launch/pr2_no_arms.launch" pass_all_args="true"/>
 </launch>
```

Example command: `roslaunch package_name file.launch`

ROS 2 is production-grade ROS

ROS 2 makes it easier to convert ROS-based prototypes into products that work effectively in real-world applications

Enterprise features:
Production quality core libraries
Enhanced security
Improved determinism
Real time communication support
Real world networking support

Layered architecture:
Multi-OS support
Industry-standard middleware layer
Improved abstraction
Unified API

Interoperability with ROS1:
ROS1 bridge enables hybrid systems

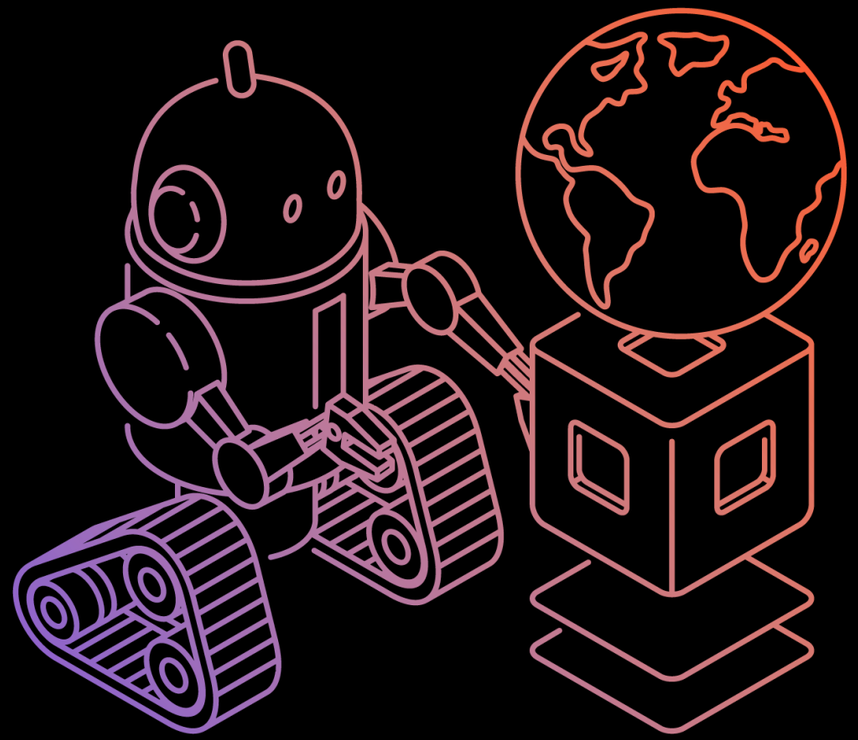


https://aws.amazon.com/blogs/robotics/ros-2-foxy-fitzroy-robot-development

Why simulation?

Building robots is HARD!

- Software and hardware development usually occurs in parallel
- Simulation allows for software to be tested independently of hardware
- Algorithm flaws and implementation issues (bugs) can be found earlier in the development process
- Software testing can be automated
- Testing on hardware can be done with much higher confidence, less time wasted



Milvus Robotics

Need

- Increase code release velocity while improving test coverage and safety for running robots in hospitals

Challenges

- Access to hospital environment was difficult, so testing was costly and time consuming
- Test coverage was limited and discovery of software defects was delayed

Solution

- Automated simulation-based regression testing in a CI/CD pipeline using AWS RoboMaker and the hospital simulation environment



SEIT UV-disinfecting robots

Results

“AWS RoboMaker simulation helped us **save money and time**—and **eliminate obstacles to scaling our testing efforts**. With RoboMaker, we can forget about infrastructure and **focus on development and testing**.

We implemented a RoboMaker simulation CI and can now **easily test our robots** in different environments, **resulting in a huge quality improvement**.”

Goker Ertunc

Co-founder & CIO
Milvus Robotics



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Gazebo 3D simulator

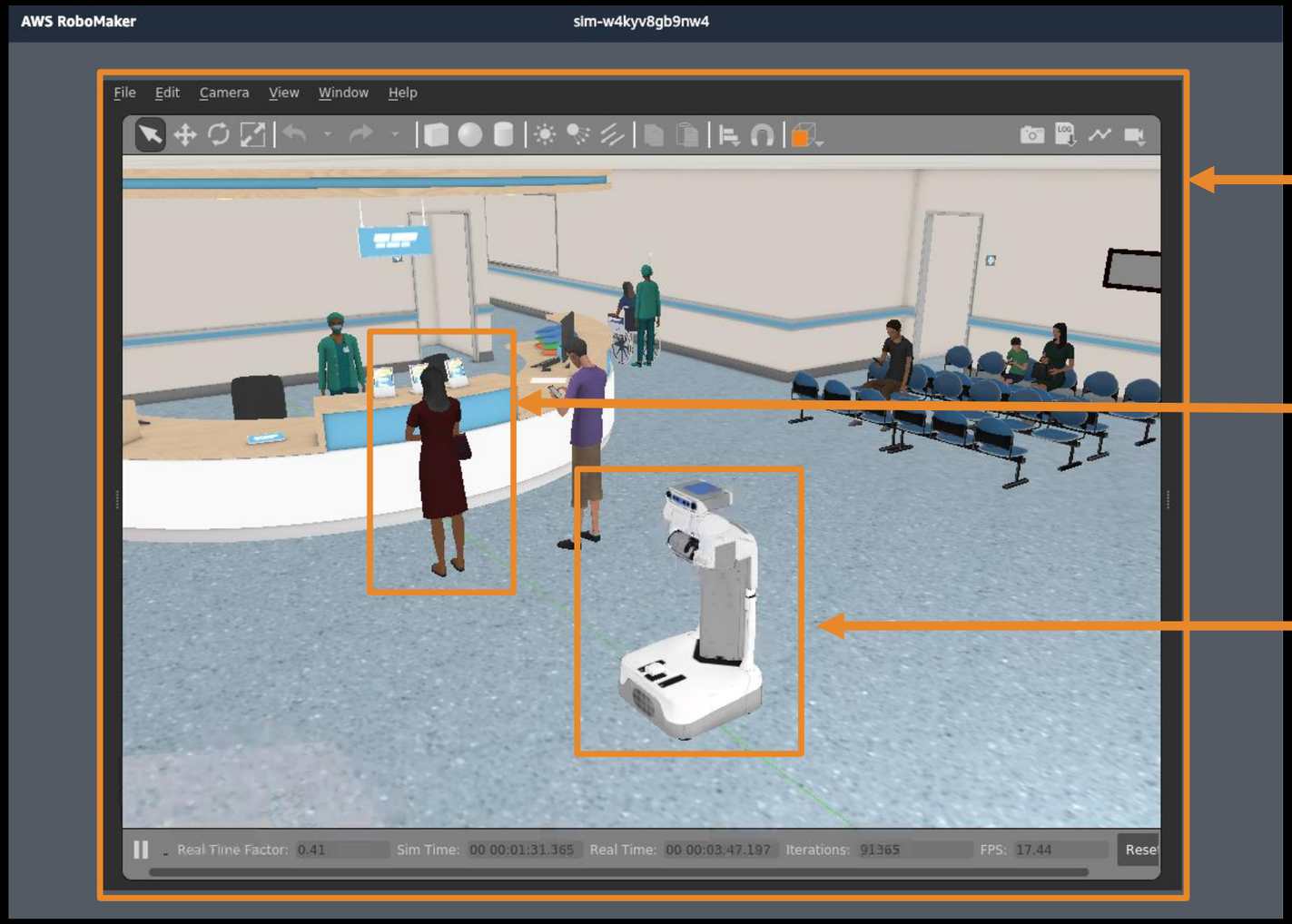


Models	SDF, URDF, OBJ STL, Collada
Physics engine	ODE, Bullet Simbody, DART
3D Rendering engine	OGRE
Robotics middleware	ROS, Player, Sockets (protobuf messages)

*Supported in AWS RoboMaker Simulation


GAZEBO

Inside a robot simulation

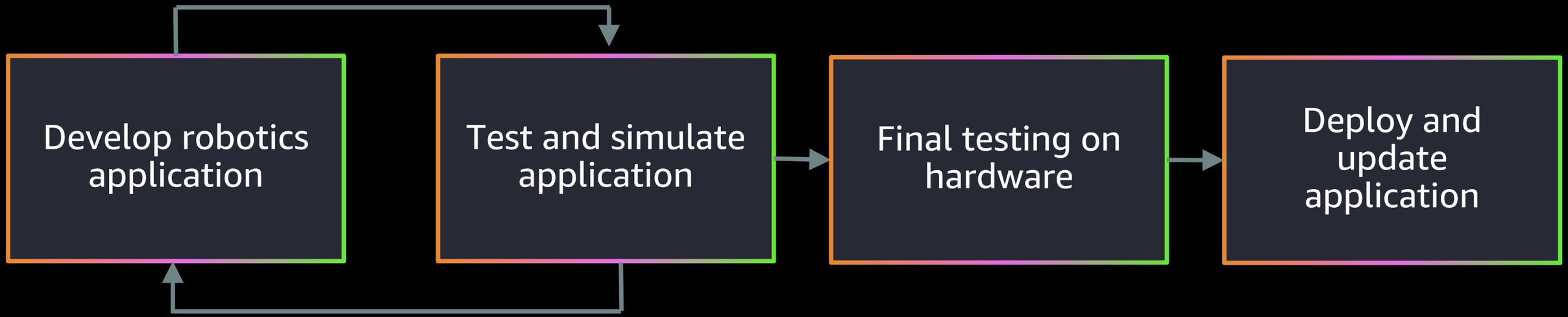


Gazebo World (SDF)

3D Model in Gazebo (SDF, with meshes—STL, OBJ, DAE, etc.)

Virtual Robot (URDF)

Robot software development cycle



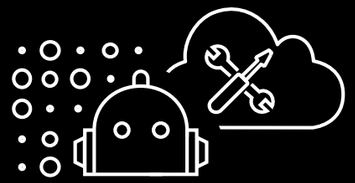
- Software development is iterative
- Testing happens at unit and system level
- Simulation at system test level as part of CI/CD flow
- Final testing on robot hardware for release candidate
- Over the air deployment to fleet

Why AWS RoboMaker?



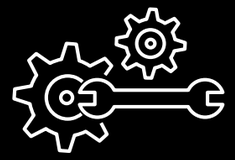
Highly scalable

Concurrent simulations at cloud-scale via a single API call



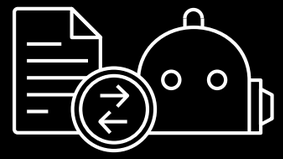
Reliable

Enterprise-grade availability with built-in health checks to ensure simulations run smoothly and reliably



Fully managed

Managed ROS and Gazebo software stack frees up engineering resources



AWS service integrations

Suite of AWS services for building end-to-end solutions

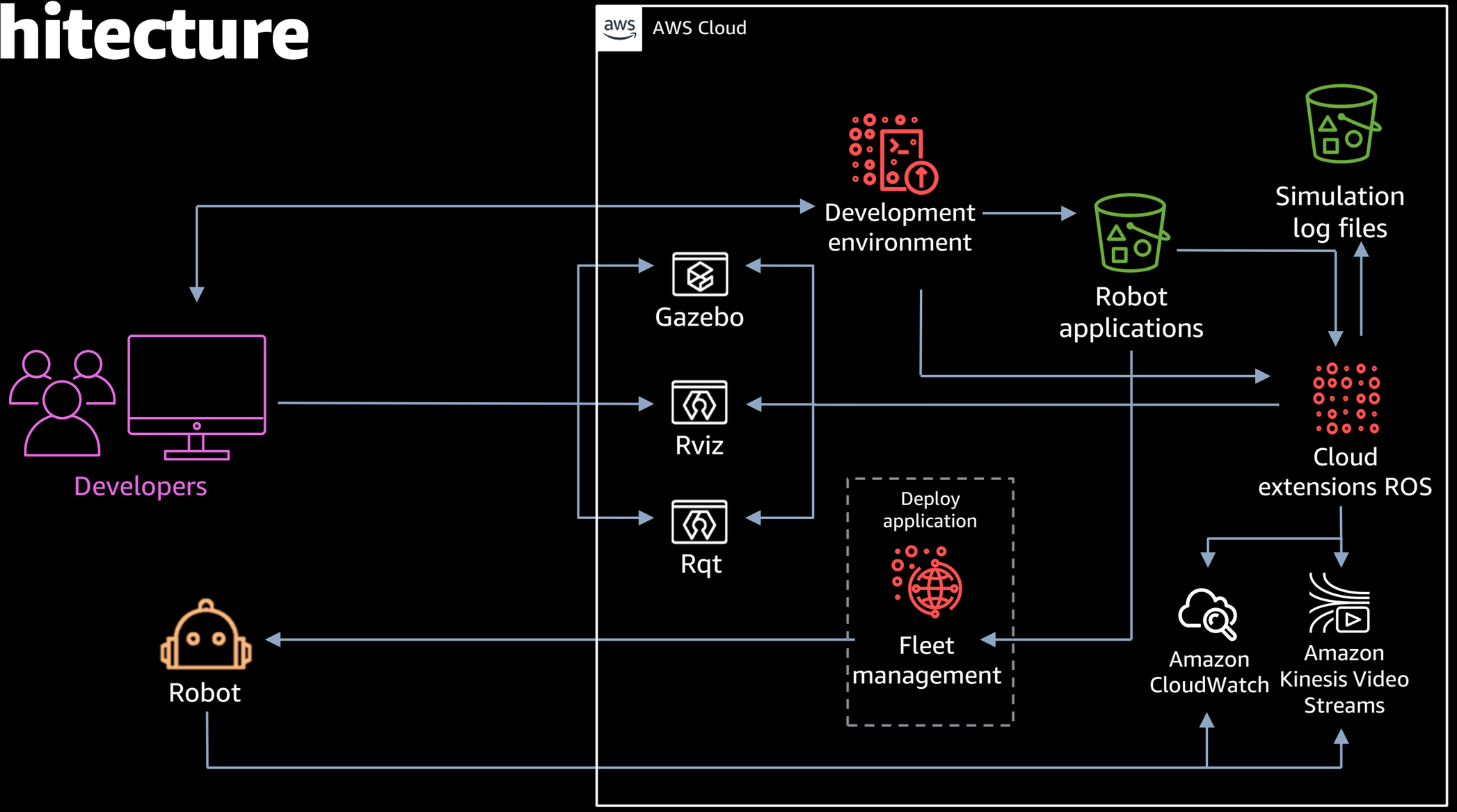


Cost effective

Pay-as-you-go pricing lowers cost of product testing

1 Simulation Unit (1 vCPU and 2GB of memory) is $0.0067 per minute

AWS RoboMaker architecture

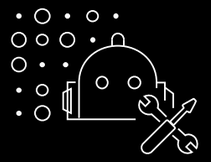


Demo: Simulating robots in a virtual Hospital Word



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



AWS ROBOMAKER

Cloud extensions for ROS

Cloud extensions written as ROS packages automatically create connections and make API calls to AWS services such as Amazon Lex, Amazon Polly, Amazon Kinesis Video Streams, Amazon Rekognition, and Amazon CloudWatch



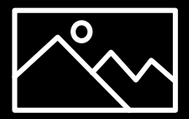
LEX
speech recognition



POLLY
speech generation



KINESIS VIDEO STREAMS
video streams



REKOGNITION
image and video analysis



CLOUDWATCH
logging and monitoring

Open-source assets

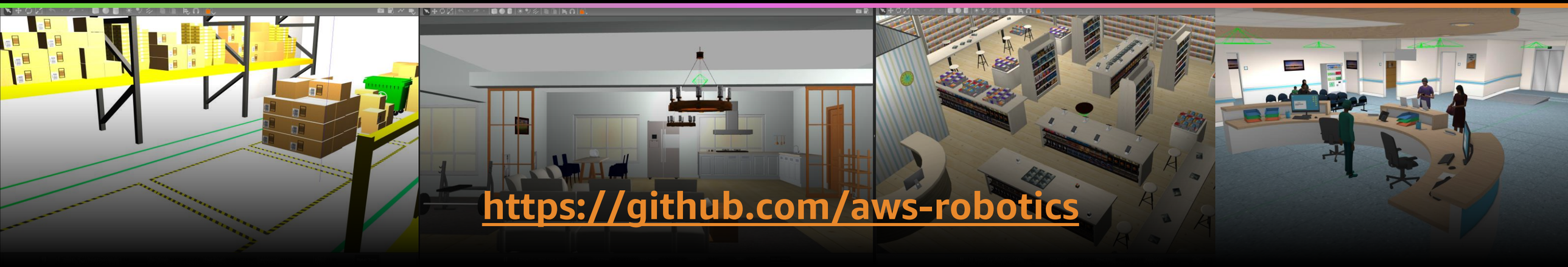
Multiple open-source 3D worlds and ROS cloud extensions are available in GitHub to use in AWS RoboMaker simulation

Cloud extensions

- ROS Bag uploader (S3)
- AWS ML services extensions (Text-to-speech, Natural language comprehension
- Amazon CloudWatch extension
- Amazon Kinesis Video extension

3D Gazebo worlds

- Hospital world
- Small warehouse world
- Small house world
- Bookstore world
- Racetrack



https://github.com/aws-robotics

Advanced robots require advanced building tools.

With the power of cloud-based simulation and open source tools, we've enabled our customers to build better robots and get them to market faster with fewer production errors.

iRobot

Jetbrain

Bastian SOLUTIONS
a TOYOTA ADVANCED LOGISTICS company

HUSARION

cheetahmobile

MiLVUS ROBOTICS

JPL
Jet Propulsion Laboratory
California Institute of Technology

TAKENAKA

Resources

Learn about ROS
https://ros.org

3D Worlds and ROS cloud extensions
https://github.com/aws-robotics

AWS RoboMaker scenario-based simulation launcher
https://github.com/aws-samples/aws-robomaker-simulation-launcher

Sample application with test node
https://github.com/aws-robotics/aws-robomaker-sample-application-cloudwatch

AWS robotics blog with detailed guides on CI/CD and more
https://aws.amazon.com/blogs/robotics

Get started with AWS RoboMaker today!
https://aws.amazon.com/robomaker

Thank you!

Matthew Hansen

Principal Solutions Architect,
Robotics

AWS

robomaker-contactus@amazon.com

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.