AWS ROBOMAKER WALL FOLLOWING ROBOT.

AWS RoboMaker is a service that makes it easy to create robotics applications at scale. AWS RoboMaker extends the Robot Operating System (ROS) framework with cloud services. This includes AWS machine learning services. It includes monitoring services. It even includes analytics services. These combine to enable a robot to do several things on its own. Stream data, navigate, communicate, comprehend, and learn. AWS RoboMaker provides a robotics application development environment. It provides a robotics simulation service, which speeds application testing. You can easily create hundreds of new worlds from templates you define using Simulation WorldForge It provides a fleet management service so you can deploy and manage applications remotely.

AWS RoboMaker is a service that allows you to quickly develop, test, and deploy robot applications. The Steps to be followed are :
Robotics Development with AWS RoboMaker
Create a ROS Development Environment
Create a Robot Application
Develop Simulation and Testing Data
Fleet Management and Deployment

1.Robotics Development with AWS RoboMaker

As a robot developer, you typically perform the following activities.Create a ROS development environment. To create a robot application, you need an environment configured for ROS development along with tools like Colcon to build and bundle the application. You'll also need tools to help you cross-compile the application for your physical robot. Using an integrated development environment makes it easier.In AWS RoboMaker, you can create an AWS Cloud9 development environment that is already configured with the tools to develop robot applications. You can also use your existing environment.Create the robot application. This is where you get to write code. Build on the foundation provided by ROS and integrate functions you find elsewhere. The application you create works with your robot hardware, provides intelligence, and works with the cloud.Develop simulation and testing data. In this stage, run your robot application in simulated environments. Collect sensor data and other performance data from the simulation. It can take many simulation tests to complete the robot programming.Deploy an application to robot fleets. When your application performs as expected, you are ready to deploy it to a robot. In AWS RoboMaker, a robot must belong to a group of robots (a fleet) in order to receive deployed software. Each virtual robot in AWS RoboMaker represents a physical robot.Monitor and update robots. Your robots are interacting in the world! Refine them by using data you collect with AWS RoboMaker cloud extensions.

2.CREATE A ROS DEVELOPMENT ENVIRONMENT.

You can create your own development environment or update an existing development environment to support AWS RoboMaker. Most developers use Ubuntu or other supported Linux variants. Other operating systems might be compatible.AWS RoboMaker provides a quick and easy way to create a development environment that is already configured for robot development.
My development environment.

3.CREATE A ROBOT APPLICATION.

After your development environment is configured, create a robot application. Build on the foundation ROS provides. ROS relies on a Computation Graph. It's a collection of concurrent processes (or nodes) that perform a task like controlling wheel motors or passing messages.You do not have to create nodes for common robotic hardware and algorithms. There are packages, which are nodes and dependent message definitions, to work with motors, lasers, actuators, lidar, and sensors of all kinds. There are also packages that consume data from other packages to create maps, find paths, and more.

4.DEVELOP SIMULATION AND TEST DATA.

Robot application developers use simulations to help refine behavior. They use simulations to test robotics algorithms and perform regression tests. Simulations use realistic scenarios and detailed virtual environments to model the world and mimic robot behavior.In AWS RoboMaker, a simulation application contains models for the robot, terrain, and assets organized in a scene. The simulation application is responsible for simulating the physical aspects of a robot such as its sensors, kinematics, and dynamics. Sensors may include cameras, lidars, and even GPS devices. Kinematics and dynamics are required to allow the robot to move its joints or wheels and interact physically, such as colliding, with objects in a simulated environment.You can generate hundreds of unique worlds from world templates using Simulation WorldForge. Each world template defines the parameters like room types and furnishing to use when generating worlds. When your world template is complete, you generate worlds using a world generation job. You can use the worlds in your simulation job or export them to use in other applications.To run a simulation, pair a robot application with a simulation application in a simulation job. The simulation job can run up to 14 days. You can restart it with a new application while it is running. You can interact with a running simulation by using Gazebo, rviz, rqt, and a terminal to interact at the command line. For example, use Gazebo to see a rendered model of the robot in the environment and use the terminal to listen or send ROS messages to your robot.The robot is unaware that it is inside a simulated environment. The simulator uses the same interfaces and data types as the robot's physical devices. This makes it possible to test the same robot software in a simulation and then deploy it to your robots.

5.FLEET MANAGEMENT AND DEPLOYMENT.

After testing is complete, you can deploy the robot application to your robots by using an AWS IoT Greengrass over-the-air update. Before you deploy your application, set up each robot to accept updates from AWS RoboMaker and communicate its status. Next, register your robots into a fleet. A fleet is a logical group of robots. When your fleet is set up, deploy your robot application. You can control the pace of deployment. You can also control what happens before and after your application launches on the robot.Information about the deployment is provided by AWS RoboMaker. Additional information specific to your robot and scenarios can be captured using AWS RoboMaker cloud extensions and custom code.