ROS-2 HUMBLE NOTES

What is ROS-2?

Its an open source framework used to build softwares for robots (despite the name including Operating System,its a library which we can use to create programs that allow us to interact with machines).

How does data move in ROS 2?

Data moves through 3 primary patterns and that is Topics, Services, and actions.

NODES:

A node is a participant in the ROS 2 graph, which uses a client library to communicate with other nodes.

Nodes can publish to named topics to deliver data to other nodes, or subscribe to named topics to get data from other nodes. They can also act as a service client to have another node perform a computation on their behalf, or as a service server to provide functionality to other nodes.

For long-running computations, a node can act as an action client to have another node perform it on their behalf, or as an action server to provide functionality to other nodes. Nodes can provide configurable parameters to change behavior during run-time.

TOPICS:

ROS 2 breaks complex systems down into many modular nodes. Topics are a vital element of the ROS graph that act as a bus for nodes to exchange messages.

A node may publish data to any number of topics and simultaneously have subscriptions to any number of topics.

SERVICES:

Services are another method of communication for nodes in the ROS graph. Services are based on a call-and-response model versus the publisher-subscriber model of topics. While topics allow nodes to subscribe to data streams and get continual updates, services only provide data when they are specifically called by a client.

ACTIONS:

Actions are one of the communication types in ROS 2 and are intended for long running tasks. They consist of three parts: a goal, feedback, and a result.

Actions are built on topics and services. Their functionality is similar to services, except actions can be canceled. They also provide steady feedback, as opposed to services which return a single response.

Actions use a client-server model, similar to the publisher-subscriber model (described in the topics tutorial). An "action client" node sends a goal to an "action server" node that acknowledges the goal and returns a stream of feedback and a result.