ROS 2 Cheat Sheet — Concise Power Commands

A compact reference for essential ROS 2 CLI commands, debugging workflows, hidden tricks, and productivity tips commonly missed by beginners and intermediate users.

ROS 2 Basics

Purpose	Command / Note
List nodes	`ros2 node list`
Node info	`ros2 node info /node_name`
List topics	`ros2 topic list`
Echo topic	`ros2 topic echo /topic`
Topic frequency	`ros2 topic hz /topic`
Topic bandwidth	`ros2 topic bw /topic`
List services	`ros2 service list`
Call service	`ros2 service call /srv std_srvs/srv/Trigger`
List actions	`ros2 action list`
Action info	`ros2 action info /action_name`

Workspace & Build

Purpose	Command / Note
Build package	`colcon build --packages-select my_pkg`
Symlink install	`colcon build --symlink-install`
Source workspace	`source install/setup.bash`
Clean build	`rm -rf build install log`
Build sequentially	`colcon build --executor sequential`

Launch & Run

Purpose	Command / Note
Run node	`ros2 run pkg node`
Launch file	`ros2 launch pkg file.launch.py`
Pass parameters	`ros2 run pkg node --ros-args -p use_sim_time:=true`

Remap topic	ros2 run pkg node --ros-args -r /cmd_vel:=/robot/cmd_vel

Parameters

Purpose	Command / Note
List params	ros2 param list
Get param	ros2 param get /node param_name
Set param	ros2 param set /node param_name value
Dump params	ros2 param dump /node
Load params	ros2 param load /node file.yaml

TF & Visualization

Purpose	Command / Note
View TF tree	ros2 run tf2_tools view_frames
Echo transform	ros2 run tf2_ros tf2_echo frame1 frame2
RViz	rviz2
Robot state publisher	ros2 run robot_state_publisher robot_state_publisher

Recording & Debugging

Purpose	Command / Note
Record bag	ros2 bag record -a
Play bag	ros2 bag play bag_folder
Record selected topics	ros2 bag record /cmd_vel /scan
Verbose logs	export RCUTILS_LOGGING_SEVERITY=DEBUG
Monitor graph	rqt_graph
GUI tools	rqt

Networking & DDS

Purpose	Command / Note
Set domain ID	export ROS_DOMAIN_ID=10
Check middleware	echo $RMW_IMPLEMENTATION
Use Cyclone DDS	export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
Discover hidden topics	ros2 topic list --include-hidden-topics

Hidden Gems (Most People Miss)

Purpose	Command / Note
Find publishers/subscribers	`ros2 topic info /topic --verbose`
See hidden nodes	`ros2 node list -a`
Inspect interfaces	`ros2 interface show pkg/msg/Type`
Generate package	`ros2 pkg create --build-type ament_python my_pkg`
See package executables	`ros2 pkg executables`
Monitor latency	`ros2 topic delay /topic`
Publish once	`ros2 topic pub --once /topic std_msgs/msg/String "{data: hello}"`
Throttle echo	`ros2 topic echo /topic --qos-reliability best_effort`
CLI YAML trick	Use YAML inline for messages/services

QoS Quick Reference

Purpose	Command / Note
Reliable	Guaranteed delivery
Best Effort	Fast, may drop packets
Transient Local	Late subscribers receive last msg
Volatile	No message persistence

Pro Tips

Purpose	Command / Note
Use tmux	Run multiple ROS nodes in panes
Use aliases	`alias cw='cd ~/ros2_ws'`
Source automatically	Add `source install/setup.bash` to `~/.bashrc`
Use Foxglove	Better visualization/debugging
Use ros2 doctor	System diagnostics checker

Tip: Most ROS 2 productivity gains come from mastering topic introspection, QoS debugging, launch remapping, and TF debugging workflows.