Devlink Reload

`devlink-reload` provides mechanism to reinit driver entities, applying `devlink-params` and `devlink-resources` new values. It also provides mechanism to activate firmware.

Reload Actions

User may select a reload action. By default `driver_reinit` action is selected.

Possible reload actions

Name	Description
`driver-reinit`	Devlink driver entities re-initialization, including applying new values to devlink entities which are used during driver load such as `devlink-params` in configuration mode `driverinit` or `devlink-resources`
`fw_activate`	Firmware activate. Activates new firmware if such image is stored and pending activation. If no limitation specified this action may involve firmware reset. If no new image pending this action will reload current firmware image.

Note that even though user asks for a specific action, the driver implementation might require to perform another action alongside with it. For example, some driver do not support driver reinitialization being performed without fw activation. Therefore, the devlink reload command returns the list of actions which were actrually performed.

Reload Limits

By default reload actions are not limited and driver implementation may include reset or downtime as needed to perform the actions.

However, some drivers support action limits, which limit the action implementation to specific constraints.

Possible reload limits

Name	Description
`no_reset`	No reset allowed, no down time allowed, no link flap and no configuration is lost.

Change Namespace

The netns option allows user to be able to move devlink instances into namespaces during devlink reload operation. By default all devlink instances are created in init_net and stay there.

example usage

```
$ devlink dev reload help
$ devlink dev reload DEV [ netns { PID | NAME | ID } ] [ action { driver_reinit | fw_activate } ] [ limit no_

# Run reload command for devlink driver entities re-initialization:
$ devlink dev reload pci/0000:82:00.0 action driver_reinit
reload_actions_performed:
  driver_reinit

# Run reload command to activate firmware:
# Note that mlx5 driver reloads the driver while activating firmware
$ devlink dev reload pci/0000:82:00.0 action fw_activate
reload_actions_performed:
  driver_reinit fw_activate
```