```
    }
```

Example: Multiple PXCs on the same underlying cross-connect configuration (classic CLI)

```
A:node-2>config>port-xc# info
----------------------------------------------
        pxc 1 create
            port 1/x1/1/c1/1
            no shutdown
        exit
        pxc 2 create
            shutdown
            port 1/x1/1/c1/1
        exit
        pxc 3 create
            shutdown
            port 1/x1/1/c1/1
        exit
    exit
```

A faceplate port that has been placed in the loopback mode for PXC use, supports only hybrid mode of operation and dot1q encapsulation. The recommendation is that the MTU value be configured to the maximum value. dot1x tunneling is enabled and cannot be changed.

The pre-set dot1q Ethernet encapsulation on the faceplate port is irrelevant from the operator's perspective and there is no need to change it. The relevant encapsulation carrying service tags defined on PXC subports and that encapsulation is configurable. For more information, see PXC sub-ports.

The following guidelines apply to a PXC configuration based on faceplate ports:

- Only unused faceplate ports (not associated with an interface or SAP) can be referenced within a PXC ID configuration.
- When the faceplate port is allocated to a PXC, it cannot be used outside of the PXC context. For example, an IP interface cannot use the faceplate port directly, or a SAP under a such port cannot be associated with an Epipe or VPLS service.

6.4 Internal PXC

With internal (or MAC-based) PXC, the egress path is cross-connected to the ingress path in the MAC chip, without the need to consume a faceplate port, as shown in Figure 21: Internal cross-connect (loopback) in a MAC chip. The number of the MAC chips on a line card varies with the line card type. The **show datapath** command shows the MAC chip related connectivity information in the datapath (forwarding complex). This information is essential for the correct configuration of the cross-connect.