La interfaz SR-IOV (Single Root I/O Virtualization) es una extensión de la especificación PCI express (PCIe) que permite a la BIOS asignar más recursos PCI a los diferentes dispositivos. Por ejemplo, si tenemos una tarjeta gráfica compatible con SRIOV ésta se subdividirá en diferentes tarjetas gráficas virtuales que se pueden pasar a diferentes máquinas virtuales de PVE como si cada una de ellas fuera una única tarjeta física. Lo mismo ocurre con las tarjetas ethernet o cualquier otra tarjeta compatible.

Para activar esta extensión en Proxmox deberemos completar 4 pasos:

- Haber completado los requisitos para poder hacer PCIPassThrough.
- Activar SR-IOV en la BIOS.
- Averiguar si el dispositivo que queremos pasar a una máquina virtual es compatible con esta extensión.
- Activarla a nivel del sistema operativo.

PCIPASSTROUGH

Simplemente sigue este hack.

BIOS

Podrás encontrar la opción en la diferentes BIOS de diferentes formas:

EN ASUS

Se encuentra en «Advanced» >> «Subsystem Settings» >> «SR-IOV Support».

EN GIGABYTE

Se encuentra con el nombre de «PCIe ARI Support».

NOTAS: Es posible que algunas placas base no tengan esta función. También es posible que algunas otras placas base si la tengan, pero no se haga mención de esta extensión en la BIOS, por lo que la tendrán activada por defecto.

DISPOSITIVO COMPATIBLE CON LA EXTENSIÓN

Para dispositivos de red, una salida vacía al comando:

```
lspci -vv -d ::200 | grep IOV
```

...nos permitiría saber rápidamente si el dispositivo a pasar es compatible o no con la extensión SR-IOV.

Si el resultado no es una salida vacía sino algo como esto:

```
Capabilities: [160] Single Root I/O Virtualization (SR-IOV)
```

Sabremos que tenemos, al menos, una tarjeta de red con funcionalidad SR-IOV. Para ver cual es, simplemente ejecutamos:

```
lspci -v -d ::200
```

Para una tarjeta ethernet PCIe de 6 puertos gigabit, nos dará una salida como esta, por cada puerto RJ-45:

```
07:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
        Subsystem: Intel Corporation 82576 Gigabit Network Connection
        Physical Slot: 117
        Flags: bus master, fast devsel, latency 0, IRQ 38, IOMMU group 28
        Memory at fc5a0000 (32-bit, non-prefetchable) [size=128K]
```

```
        Memory at fc580000 (32-bit, non-prefetchable) [size=128K]
        I/O ports at d020 [disabled] [size=32]
        Memory at fc644000 (32-bit, non-prefetchable) [size=16K]
        Expansion ROM at fc560000 [disabled] [size=128K]
        Capabilities: [40] Power Management version 3
        Capabilities: [50] MSI: Enable- Count=1/1 Maskable+ 64bit+
        Capabilities: [70] MSI-X: Enable+ Count=10 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [140] Device Serial Number 00-e0-ed-ff-ff-19-7f-e4
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [160] Single Root I/O Virtualization (SR-IOV)
        Kernel driver in use: igb
        Kernel modules: igb
```

Podemos ver que la tarjeta es compatible con SR-IOV porque tiene la capability 160 y que esta es administrada por el módulo **igb**. Además, la tarjeta tiene que contar con la capability 150 (ARI, Alternative Routing ID) porque es esa capability, la que junto a la 160 ofrecen la funcionalidad SR-IOV. Deben estar las dos.

Para hilar un poco más fino y ver cuantas «virtual functions» tiene por cada uno de esos puertos ethernet, podemos ejecutar como root:

`lspci -vv -d ::200`

La salida será algo como esto, por cada uno de los puertos RJ-45:

```
07:00.1 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
        Subsystem: Intel Corporation 82576 Gigabit Network Connection
        Physical Slot: 117
        Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
        Latency: 0, Cache Line Size: 64 bytes
        Interrupt: pin B routed to IRQ 32
        IOMMU group: 29
        Region 0: Memory at fc540000 (32-bit, non-prefetchable) [size=128K]
        Region 1: Memory at fc520000 (32-bit, non-prefetchable) [size=128K]
        Region 2: I/O ports at d000 [disabled] [size=32]
        Region 3: Memory at fc640000 (32-bit, non-prefetchable) [size=16K]
        Expansion ROM at fc500000 [disabled] [size=128K]
        Capabilities: [40] Power Management version 3
          Flags: PMEClk- DSI+ D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold-)
          Status: D0 NoSoftRst- PME-Enable- DSel=0 DScale=1 PME-
        Capabilities: [50] MSI: Enable- Count=1/1 Maskable+ 64bit+
          Address: 0000000000000000 Data: 0000
          Masking: 00000000 Pending: 00000000
        Capabilities: [70] MSI-X: Enable+ Count=10 Masked-
          Vector table: BAR=3 offset=00000000
          PBA: BAR=3 offset=00002000
        Capabilities: [a0] Express (v2) Endpoint, MSI 00
          DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s <512ns, L1 <64us
          ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset+ SlotPowerLimit 0.000W
          DevCtl: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq+
            RlxdOrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+ FLReset-
            MaxPayload 256 bytes, MaxReadReq 512 bytes
          DevSta: CorrErr+ NonFatalErr- FatalErr- UnsupReq+ AuxPwr- TransPend-
          LnkCap: Port #5, Speed 2.5GT/s, Width x4, ASPM L0s L1, Exit Latency L0s <4us, L1 <64us
            ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp-
          LnkCtl: ASPM Disabled; RCB 64 bytes, Disabled- CommClk-
            ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
          LnkSta: Speed 2.5GT/s (ok), Width x4 (ok)
            TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
          DevCap2: Completion Timeout: Range ABCD, TimeoutDis+ NROPrPrP- LTR-
            10BitTagComp- 10BitTagReq- OBFF Not Supported, ExtFmt- EETLPPrefix-
            EmergencyPowerReduction Not Supported, EmergencyPowerReductionInit-
            FRS- TPHComp- ExtTPHComp-
            AtomicOpsCap: 32bit- 64bit- 128bitCAS-
          DevCtl2: Completion Timeout: 16ms to 55ms, TimeoutDis- LTR- OBFF Disabled,
```

```
            AtomicOpsCtl: ReqEn-
          LnkSta2: Current De-emphasis Level: -6dB, EqualizationComplete- EqualizationPhase1-
            EqualizationPhase2- EqualizationPhase3- LinkEqualizationRequest-
            Retimer- 2Retimers- CrosslinkRes: unsupported
        Capabilities: [100 v1] Advanced Error Reporting
          UESta: DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
          UEMsk: DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
          UESvrt: DLP+ SDES- TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+ ECRC- UnsupReq- ACSViol-
          CESta: RxErr- BadTLP- BadDLLP- Rollover- Timeout- AdvNonFatalErr+
          CEMsk: RxErr- BadTLP- BadDLLP- Rollover- Timeout- AdvNonFatalErr+
          AERCap: First Error Pointer: 00, ECRCGenCap- ECRCGenEn- ECRCChkCap- ECRCChkEn-
            MultHdrRecCap- MultHdrRecEn- TLPPfxPres- HdrLogCap-
          HeaderLog: 00000000 00000000 00000000 00000000
        Capabilities: [140 v1] Device Serial Number 00-e0-ed-ff-ff-19-7f-e4
        Capabilities: [150 v1] Alternative Routing-ID Interpretation (ARI)
          ARICap: MFVC- ACS-, Next Function: 0
          ARICtl: MFVC- ACS-, Function Group: 0
        Capabilities: [160 v1] Single Root I/O Virtualization (SR-IOV)
          IOVCap: Migration-, Interrupt Message Number: 000
          IOVCtl: Enable- Migration- Interrupt- MSE- ARIHierarchy-
          IOVSta: Migration-
          Initial VFs: 8, Total VFs: 8, Number of VFs: 0, Function Dependency Link: 01
          VF offset: 128, stride: 2, Device ID: 10ca
          Supported Page Size: 00000553, System Page Size: 00000001
          Region 0: Memory at 00000000fc5e0000 (64-bit, non-prefetchable)
          Region 3: Memory at 00000000fc5c0000 (64-bit, non-prefetchable)
          VF Migration: offset: 00000000, BIR: 0
        Kernel driver in use: igb
        Kernel modules: igb
```

Si volvemos a mirar más en detalle la capability 160 podemos observar que tiene 8 virtual functions. Es decir, permite virtualizar 8 adaptadores extra por cada uno de los puertos RJ45.

Si listamos las interfaces de la tarjeta de red con

`ip a`

…antes de que activemos SR-IOV podremos ver la siguiente salida:

xxx

Más adelante volveremos a ejecutar **ip a**, luego de activar SR-IOV, y veremos como aparecen nuevas interfaces virtuales.

ACTIVAR SR-IOV A NIVEL DE SISTEMA OPERATIVO

Para activar la extensión a nivel del sistema operativo y que este puede utilizar todas las interfaces virtuales disponibles en el adaptador de red, ejecutamos:

```
echo "options igb max_vfs=8" > /etc/modprobe.d/sriov.igb.conf
update-initramfs -u -k all
```

Como vemos, le estamos indicando al sistema operativo que el módulo igb pueda activar el control de 8 funciones virtuales por cada adaptador que administre. Si la tarjeta de red fuera administrada por otro módulo del kernel o tuviera más de 8 funciones virtuales por cada puerto ethernet, tendríamos que hacer los cambios correspondientes en el comando de arriba.

Pero antes de reiniciar el sistema, vamos a ejecutar ip a para ver las diferentes interfaces que este adaptador de red proporciona al sistema operativo sin la extensión SR-IOV:

xxx

Ahora si, si volvemos a ejecutar ip a después de haber reiniciado el sistema operativo, podemos encontrar la siguiente salida, por cada uno de

los puertos ethernet:

```
eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether ea:f9:39:3e:59:b4 brd ff:ff:ff:ff:ff:ff permaddr b2:bd:b0:d3:fb:f6
      altname enp7s0f0v0
eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether ca:68:18:54:c4:8c brd ff:ff:ff:ff:ff:ff permaddr ee:59:3c:a7:66:c7
      altname enp7s0f0v2
eth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether 36:b1:ac:e2:c1:43 brd ff:ff:ff:ff:ff:ff permaddr 56:e3:54:d9:e2:ec
      altname enp7s0f0v1
eth3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether 7e:73:37:94:4f:71 brd ff:ff:ff:ff:ff:ff permaddr 36:f4:0e:5f:16:49
      altname enp7s0f0v3
eth4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether c6:29:78:97:e2:22 brd ff:ff:ff:ff:ff:ff permaddr 2a:0a:77:47:fb:26
      altname enp7s0f0v4
eth5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether 1a:50:92:3f:14:9e brd ff:ff:ff:ff:ff:ff permaddr ea:5a:8e:da:a0:da
      altname enp7s0f0v5
eth6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether ae:55:28:09:cc:b0 brd ff:ff:ff:ff:ff:ff permaddr 2a:ec:2d:48:ee:db
      altname enp7s0f0v6
eth7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
      link/ether 00:e0:ed:19:7f:e4 brd ff:ff:ff:ff:ff:ff
      altname enp7s0f0
```

De esto podemos observar que eth7 (con alias enp7s0f0) es la «interfaz madre» de la que se desprenden las otras 8 interfaces virtuales: enp7s0f0v0 a enp7s0f0v7 (con la terminación v#). Esto es así por cada uno de los puertos Ethernet. No he puesto toda la salida completa en este hack porque sino se haría muy extenso.

Ahora , si ejecutamos:

`lspci -v -d ::200`

Podemos ver la siguiente salida:

```
07:00.0 Ethernet controller: Intel Corporation 82576 Gigabit Network Connection (rev 01)
        Subsystem: Intel Corporation 82576 Gigabit Network Connection
        Physical Slot: 117
        Flags: bus master, fast devsel, latency 0, IRQ 38, IOMMU group 28
        Memory at fc5a0000 (32-bit, non-prefetchable) [size=128K]
        Memory at fc580000 (32-bit, non-prefetchable) [size=128K]
        I/O ports at d020 [disabled] [size=32]
        Memory at fc644000 (32-bit, non-prefetchable) [size=16K]
        Expansion ROM at fc560000 [disabled] [size=128K]
        Capabilities: [40] Power Management version 3
        Capabilities: [50] MSI: Enable- Count=1/1 Maskable+ 64bit+
        Capabilities: [70] MSI-X: Enable+ Count=10 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [140] Device Serial Number 00-e0-ed-ff-ff-19-7f-e4
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Capabilities: [160] Single Root I/O Virtualization (SR-IOV)
        Kernel driver in use: igb
        Kernel modules: igb

07:10.0 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
```

```
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.1 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.2 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.3 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.4 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.5 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.6 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
```

```
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf

07:10.7 Ethernet controller: Intel Corporation 82576 Virtual Function (rev 01)
        Subsystem: Intel Corporation 82576 Virtual Function
        Flags: bus master, fast devsel, latency 0, IOMMU group 43
        Memory at fc620000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Memory at fc600000 (64-bit, non-prefetchable) [virtual] [size=16K]
        Capabilities: [70] MSI-X: Enable+ Count=3 Masked-
        Capabilities: [a0] Express Endpoint, MSI 00
        Capabilities: [100] Advanced Error Reporting
        Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
        Kernel driver in use: igbvf
        Kernel modules: igbvf
```

El primer dispositivo es uno de los puertos ethernet padre, y cada uno de los 8 siguientes es un dispositivo virtual. Podemos saberlo porque el dispositivo PCIe real está controlado por el módulo igb, mientras que los otros por igbvf. Todos los virtuales son controlados por el mismo módulo que el dispositivo físico, pero con «vf» (virtual Function) al final.

Para pasarlo a la máquina virtual haremos lo mismo que hacemos con PCIPassThrough pero, una vez dentro de la máquina virtual, para que ésta reconozca el dispositivo es probable que haga falta descargar los drivers desde la página del fabricante porque, al menos en el caso de máquinas virtuales de Windows, el sistema operativo reconocerá el dispositivo padre e instalará los controladores correspondientes pero no reconocerá los sub-dispositivos con virtual functions, de forma que tocará ir a Internet a descargar los drivers completos.