

1B. [DATADA] Instalação e configuração completa do Whitebox Gen 1.1 (Fedora 26)

Instalação

Instalar OS: Fedora 26 Server

- Keyboard: Portuguese (Brazil)
- Language: English (United States)
- Installation Source: Closest Mirror (basta aguardar alguns minutos que opção será selecionada automaticamente)
- Software Selection: Fedora Server Edition
- Installation Destination: Automatic partitioning
- Network & Host Name:
 - Full name: whitebox
 - User name: whitebox
 - Password: fibre42

Make this user administrator

Configuração

Atualizações Automáticas de Segurança:

Instalar o dnf-automatic

```
sudo dnf -y install dnf-automatic
```

Edite o arquivo /etc/dnf/automatic.conf e altere o parâmetro upgrade_type

```
upgrade_type = security
```

Inicializar o dnf-automatic

```
sudo systemctl enable dnf-automatic-install.timer && sudo systemctl start dnf-automatic-install.timer
```

Tuning:

Irqbalance:

Para restringir que as interrupções de hardware sejam tratadas pelos cores que serão alocados para o DPDK, edite o arquivo /etc/sysconfig/irqbalance e descomente parâmetro IRQBALANCE_BANNED_CPUS alterando para o valor abaixo:

```
IRQBALANCE_BANNED_CPUS="EE"
```



Limitando os núcleos utilizados para interrupções de hardware

Core:

```
7 6 5 4 3 2 1 0  
1 1 1 0 1 1 1 0 = 0xee (apenas core0 a core4 serão utilizados)
```

1 = core habilitado
0 = core desabilitado

Reiniciar o irqbalance:

```
sudo systemctl restart irqbalance
```

Tuned:

Instalar Tuned

```
sudo dnf -y install tuned
```

Criar pasta para o perfil de tuning do whitebox

```
sudo mkdir /etc/tuned/whitebox
```

Criar o arquivo /etc/tuned/whitebox/tuned.conf com o conteúdo abaixo:

```
[main]
include=network-latency

[bootloader]
cmdline = audit=0 selinux=0 isolcpus=1-3,5-7 nosoftlockup mce=ignore_ce idle=poll nohz_full=1-3,5-7
hugepages=2048

[sysctl]
kernel.nmi_watchdog=0
vm.stat_interval=10
vm.nr_hugepages=2048

[sysfs]
/sys/bus/workqueue/devices/writeback/cpumask = 17
/sys/devices/virtual/workqueue/cpumask = 17
```

Habilitar o serviço tuned e ativar o perfil whitebox

```
sudo systemctl enable tuned
sudo tuned-adm profile whitebox
```

Desabilitar Selinux:

```
sudo setenforce 0
```

Driverctl e DPDK:

instalar dependências:

```
sudo dnf -y install driverctl dpdk
```

Defina as interfaces que usarão o driver dpdk com o driverctl:

```
sudo driverctl set-override 0000:00:14.0 uio_pci_generic
sudo driverctl set-override 0000:00:14.1 uio_pci_generic
sudo driverctl set-override 0000:00:14.2 uio_pci_generic
sudo driverctl set-override 0000:00:14.3 uio_pci_generic
sudo driverctl set-override 0000:05:00.0 uio_pci_generic
sudo driverctl set-override 0000:05:00.1 uio_pci_generic
```

OpenvSwitch:

Instalar pacotes

```
sudo dnf install -y openvswitch
```

Inicie o modulo tun e adicione ele para ser inicializado automaticamente:

```
sudo modprobe tun
echo "tun" | sudo tee /etc/modules-load.d/tun.conf
```

Inicie o serviço do openvswitch:

```
sudo systemctl enable openvswitch.service && sudo systemctl start openvswitch.service
```

Configurações para inicialização do dpdk no openvswitch e alocação de memória:

```
sudo ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-init=true
sudo ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-socket-mem="4096,0"
sudo ovs-vsctl --no-wait set Open_vSwitch . other_config:pmd-cpu-mask=EE
sudo systemctl restart openvswitch.service
```



Habilitando os núcleos para dpdk

Core:

```
7 6 5 4 3 2 1 0
1 1 1 0 1 1 1 0 = 0xee (core1-3 + core5-7 reservados)
```

1 = core habilitado
0 = core desabilitado

Configurando o openvswitch para inserir controladores, bridge, filas e interface:

Segue abaixo dois modelos de configuração. Um para criar um switch virtual de 6 portas e outro para criar dois switches virtuais com 4 portas (3 portas físicas e 1 lógica)

Switch Virtual de 6 portas

Bridge

```
sudo ovs-vsctl add-br br-dpdk0 -- set bridge br-dpdk0 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
```

Ports

```
sudo ovs-vsctl add-port br-dpdk0 dpdk0 -- set Interface dpdk0 type=dpdk mtu_request=9600 ofport_request=1
options:dpdk-devargs=0000:00:14.0

sudo ovs-vsctl add-port br-dpdk0 dpdk1 -- set Interface dpdk1 type=dpdk mtu_request=9600 ofport_request=2
options:dpdk-devargs=0000:00:14.1

sudo ovs-vsctl add-port br-dpdk0 dpdk2 -- set Interface dpdk2 type=dpdk mtu_request=9600 ofport_request=3
options:dpdk-devargs=0000:00:14.2

sudo ovs-vsctl add-port br-dpdk0 dpdk3 -- set Interface dpdk3 type=dpdk mtu_request=9600 ofport_request=4
options:dpdk-devargs=0000:00:14.3

sudo ovs-vsctl add-port br-dpdk0 dpdk4 -- set Interface dpdk4 type=dpdk mtu_request=9600 ofport_request=5
options:dpdk-devargs=0000:05:00.0

sudo ovs-vsctl add-port br-dpdk0 dpdk5 -- set Interface dpdk5 type=dpdk mtu_request=9600 ofport_request=6
options:dpdk-devargs=0000:05:00.1
```

Dois Switches Virtuais de 4 portas

Bridge

```
sudo ovs-vsctl add-br br-dpdk0 -- set bridge br-dpdk0 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
sudo ovs-vsctl add-br br-dpdk1 -- set bridge br-dpdk1 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
```

Portas da Bridge br-dpdk0:

```
sudo ovs-vsctl add-port br-dpdk0 dpdk0 -- set Interface dpdk0 type=dpdk mtu_request=9600 ofport_request=1
options:dpdk-devargs=0000:00:14.0

sudo ovs-vsctl add-port br-dpdk0 dpdk2 -- set Interface dpdk2 type=dpdk mtu_request=9600 ofport_request=2
options:dpdk-devargs=0000:00:14.2

sudo ovs-vsctl add-port br-dpdk0 dpdk4 -- set Interface dpdk4 type=dpdk mtu_request=9600 ofport_request=3
options:dpdk-devargs=0000:05:00.0

sudo ovs-vsctl add-port br-dpdk0 patch0 -- set Interface patch0 type=patch options:peer=patch1 mtu_request=9600
ofport_request=4
```

Portas da Bridge br-dpdk1:

```
sudo ovs-vsctl add-port br-dpdk1 dpdk1 -- set Interface dpdk1 type=dpdk mtu_request=9600 ofport_request=1
options:dpdk-devargs=0000:00:14.1

sudo ovs-vsctl add-port br-dpdk1 dpdk3 -- set Interface dpdk3 type=dpdk mtu_request=9600 ofport_request=2
options:dpdk-devargs=0000:00:14.3

sudo ovs-vsctl add-port br-dpdk1 dpdk5 -- set Interface dpdk5 type=dpdk mtu_request=9600 ofport_request=3
options:dpdk-devargs=0000:05:00.1

sudo ovs-vsctl add-port br-dpdk1 patch1 -- set Interface patch1 type=patch options:peer=patch0 mtu_request=9600
ofport_request=4
```

Configuração do Controlador

```
sudo ovs-vsctl set-controller br-dpdk0 tcp:<ip controller>:6633
sudo ovs-vsctl set-controller br-dpdk1 tcp:<ip controller>:6633
```

Reboot o whitebox:

```
sudo reboot
```

Após reinicialização verificar se o openvswitch iniciou corretamente:

```
sudo systemctl status openvswitch.service  
sudo ovs-ofctl show br-dpdk0
```