Installazione automatica di server su BIOS UEFI

Diego Michelotto (diego.michelotto@cnaf.infn.it)

Indice

- BIOS UEFI
  - Perché
  - Dettagli
- Unattended Installation
  - DHCP
  - PXE – Boot Loader
- Unattended Installation con Foreman
  - Boot Loader template
    - Network interface
  - Partition table
  - First boot dopo installazione
- BIOS UEFI e oVirt/VMware
- SECURE BOOT
- Conclusioni

BIOS UEFI - Perché

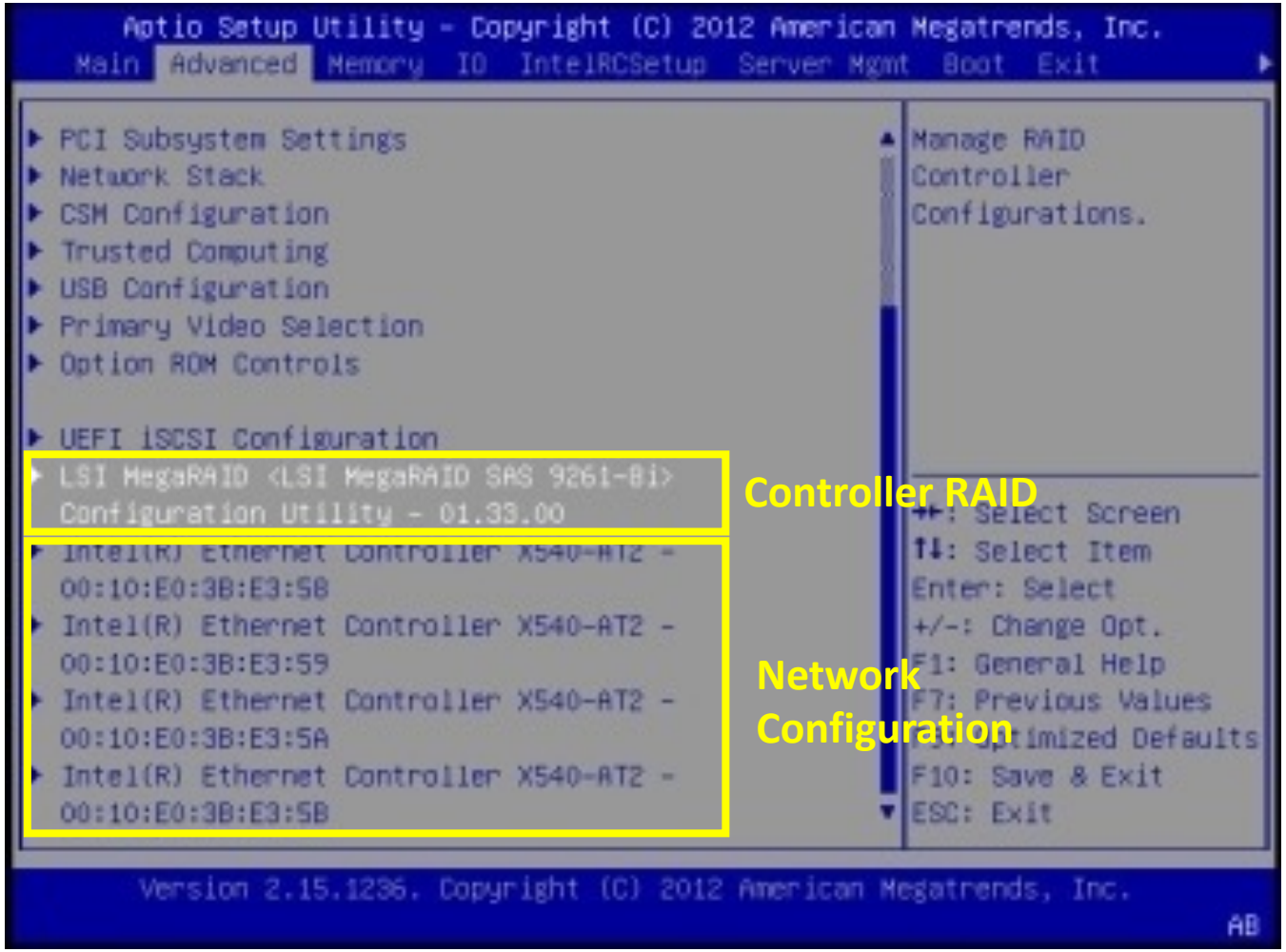
- **Perché abbiamo bisogno di UEFI**? Ho sempre usato BIOS legacy senza problemi…
  - Alcune schede di rete per poter fare **boot via PXE** devono per forza esser configurate con UEFI
    - Broadcom BCM57414 2x25Gb/s
  - Alcune schede di rete per poter **configurare FEC mode** devono per forza essere configurate con UEFI
    - Broadcom BCM57414 2x25Gb/s
  - Alcuni controller RAID per poter **configurare i livelli di RAID e i volumi** devono per forza essere configurati con UEFI
    - Broadcom / LSI MegaRAID SAS 9361-16i

BIOS UEFI - Perché

- C’è un **alternativa**?
  - Vanno installati manualmente **i tool specifici** delle schede e fatta la configurazione richiesta
    - **Necessitano di un Sistema Operativo** → Se la rete non funziona il SO va installato a mano
    - Potrebbe trattarsi di **configurazioni da fare per installare il SO** come configurazioni RAID
  - Questa alternativa non ci va bene, noi vogliamo fare delle **Unattended Installation**, eventualmente fare configurazioni particolari ma via IPMI/BMC/RedFish

BIOS UEFI - Perché

- **Vantaggi**?
  - **Tutte le configurazioni** si possono fare dall'interfaccia **Setup Utility** del sistema



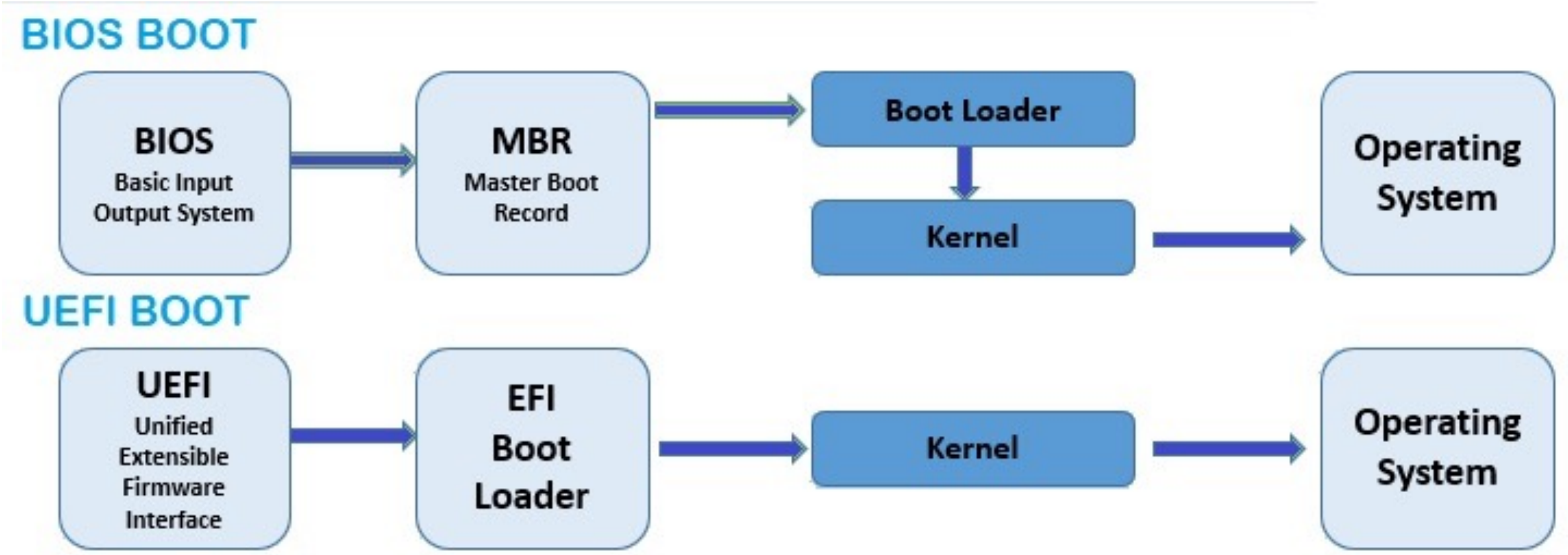
BIOS UEFI - Dettagli

- **Unified Extensible Firmware Interface**
  - **Brevetto di Intel del 2003**, sviluppato poi da **Unified EFI Forum**
- **BIOS** è **Basic Input-Output System** incorporato in UEFI
- **Retrocompatibile** grazie a **Boot Legacy** (MBR) e CSM (Compatibility Support Module)
- Basato su **GPT** invece di MBR
  - Necessità che il disco abbia GUID Partition Table
  - La prima partizione (**ESP** – EFI Sistem Partition) del disco deve essere di tipo EFI montata in /boot/efi
    - **Contiene il bootloader** grubx64.efi o shimx64.efi
  - La seconda partizione del disco deve essere di tipo ext4 o xfs montata in /boot
    - **Contiene** le varie versioni delle immagini **initrd** e dei **kernel** e la configurazione del bootloader.

BIOS UEFI - Dettagli

- **Boot Manager integrato**
  - Gestisce più dischi di avvio, Memorizza l'ultimo usato nella lista di boot
- **Unica interfaccia** per tutti i **firmware** dei vari componenti **hardware**
  - I produttori di hardware scrivono dei **driver UEFI**
- Funzionalità di rete anche senza SO
- Shell UEFI
  - Gestione Boot, diagnosi e risoluzioni problemi
- Supporta esecuzione moduli firmati **SECURE BOOT**
  - Prima di lanciare il bootloader (shimx64.efi) e il kernel, vengono verificate le firme con il db delle firme del firmware UEFI
  - Spesso in combinazione con TPM (Trusted Platform Module)
- **Svantaggi**
  - Supporta **solo sistemi 64bit**
  - Avendo il supporto alla rete può essere **soggetto a virus**

BIOS UEFI


BIOS BOOT
BIOS
Basic Input Output System
MBR
Master Boot Record
Boot Loader
Kernel
Operating System
UEFI BOOT
UEFI
Unified Extensible Firmware Interface
EFI Boot Loader
Kernel
Operating System

Unattended Installation

- Installazione automatica di sistemi:
  - Utilizzo di sistemi di **installazione via rete**
  - **Ricette per l'installazione**:
    - Partizionamento disco
    - Installazione SO
    - Installazione software base (Puppet)
- Vantaggi:
  - Indispensabile quando si installano **grandi numeri** di macchine
  - **Riproducibilità** delle installazioni

- Diversi modi:
  - **PXE boot**
    - **DHCP, TFTP e HTTP**
  - HTTP boot (solo UEFI)

Unattended Installation - DHCP

- Configurazione del DHCP
  - Assegnazione IP in base MAC address
  - Assegnazione next-server e bootloader
  - Assegnazione hostname

UEFI Grub2

```
host servername.infn.it {
  dynamic;
  hardware ethernet 00:16:3e:XX:YY:ZZ;
  fixed-address 192.168.1.2;
        supersede server.filename = "grub2/grubx64.efi";
        supersede server.next-server = c0:a8:01:01;
        supersede host-name = servername.infn.it";
}
```

UEFI Grub2
SECURE BOOT

```
host servername.infn.it {
  dynamic;
  hardware ethernet 00:16:3e:XX:YY:ZZ;
  fixed-address 192.168.1.2;
        supersede server.filename = "grub2/shimx64.efi";
        supersede server.next-server = c0:a8:01:01;
        supersede host-name = servername.infn.it";
}
```

Unattended Installation - PXE – Boot Loader

- Diversi boot loader
  - PXELinux
  - Grub
  - **Grub2**
  - iPXE
- Recuperato via TFTP sul next-server con la configurazione

```
menuentry 'Kickstart default PXEGrub2’ {
  linuxefi boot/vmlinuz inst.ks=http://serverhttp.infn.it/kickstart BOOTIF=01-00-16-3e-XX-YY-
ZZ kssendmac ks.sendmac inst.ks.sendmac
ip=192.168.1.2::192.168.1.1:255.255.255.0:servename.infn.it:enp1s0:none nameserver=8.8.8.8
  initrdefi boot/initrd.img
}
```

- Attenzione al nome dell’interfaccia

Unattended Installation con Foreman

- **Foreman** is a complete **lifecycle management tool** for physical and virtual servers.
  - Easily automate repetitive tasks, quickly deploy applications, and proactively manage servers, on-premise or in the cloud. Ref. https://theforeman.org/
- Uso di **Foreman Smart Proxy** per interagire con
  - **DHCP** → imposta dinamicamente il **lease** della macchina
  - **TFTP** → scarica **initrd e vmlinuz** per il SO da installare e configura **la entry di PXE** per l'installazione
- Gestisce **provisioning template community** o **custom**
  - Kickstart → RedHat e derivate
  - Preseed → Debian o derivate
  - Ecc
- Gestisce le **partition table** da integrare nel provisioning template
- Possiblità di fare bulk installation tremite uso di CLI o API

FOREMAN

Unattended Installation con Foreman – Boot Loader Template

- Per installazioni **UEFI** usiamo **Grub2** come bootloader
- Usiamo il **template default** di Foreman
- Modificata linea **linuxcmd** per permettere la configurazione della **console seriale** per vedere la console via ipmitool che viene abilitata tramite il parametro **serial_console** passato come **global parameter**

```
menuentry '<%= template_name %>' {
  <%= linuxcmd %> <%= @kernel %> inst.ks=<%= foreman_url('provision') %> <%= pxe_kernel_options
%> <%= snippet("kickstart_kernel_options").strip %> <%= host_param('serial_console') ?
'console=ttyS0,115200n8' : '' %>
  <%= initrdcmd %> <%= @initrd %>
}
```

- https://github.com/theforeman/community-templates/blob/develop/provisioning_templates/PXEGrub2/kickstart_default_pxegrub2.erb

Unattended Installation con Foreman – Partition table

- La **prima partizione** del disco deve essere la partizione di **tipo efi** montata in **/boot/efi**
- La **seconda partizione** deve essere montata in **/boot**
- Il resto dipende dalla destinazione d'uso, in questo caso vengono creati due logical volume in lvm, uno per swap da 4GB e il resto per /
- Es per RedHat e derivate

```
# Partition clearing information
clearpart --all --initlabel
# logical volumes creation
part /boot/efi --fstype=efi --size=200
part /boot --fstype=ext4 --size=512
part pv.008002 --grow --size=1
volgroup vg_vol01 --pesize=4096 pv.008002
logvol / --fstype=ext4 --name=lv_root --vgname=vg_vol01 --grow --size=10000
logvol swap --name=lv_swap --vgname=vg_vol01 --grow --size=1024 --maxsize=4096
```

Unattended Installation con Foreman – Partition table

- Esempio con RAID1 software per RedHat e derivate

```
clearpart --drives=sda,sdb --all -initlabel

part raid.10   --size=200    --ondisk=sda --fstype=efi
part raid.11   --size=1024   --ondisk=sda --fstype=ext4
part raid.12   --size=101376 --ondisk=sda -grow

part raid.20   --size=200    --ondisk=sdb --fstype=efi
part raid.21   --size=1024   --ondisk=sdb --fstype=ext4
part raid.22   --size=101376 --ondisk=sdb -grow

raid /boot/efi --level=RAID1 --device=md0 --fstype=efi raid.10 raid.20
raid /boot --level=RAID1 --device=md1 --fstype=ext4 raid.11 raid.21
raid pv.01 --level=RAID1 --device=md2 raid.12 raid.22

volgroup vg_vol01 pv.01
logvol / --fstype=ext4 --name=lv_root --vgname=vg_vol01 --grow --size=10000 --
maxsize=256000
logvol swap --name=lv_swap --vgname=vg_vol01 --grow --size=1024 --maxsize=8192
```

Unattended Installation con Foreman – First Boot

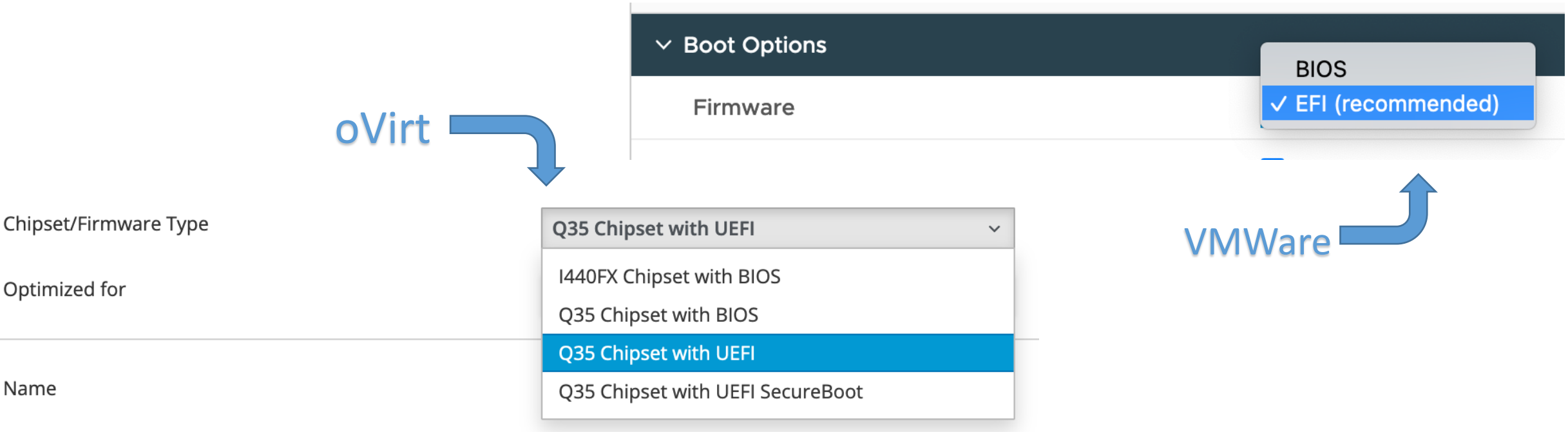
- Usiamo **default PXEGrub2 global template**
  - Problema con sistemi operativi **RockyLinux** e **AlmaLinux** → PXE non trova il bootloader perché i due sistemi operativi non sono specificati nel template **pxegrub2_chainload**

```
<%
  paths = [
    '/EFI/fedora/shim.efi',
    '/EFI/fedora/grubx64.efi',
    '/EFI/redhat/shim.efi',
    '/EFI/redhat/grubx64.efi',
    '/EFI/centos/shim.efi',
    '/EFI/centos/grubx64.efi',
    '/EFI/rocky/shim.efi',
    '/EFI/rocky/grubx64.efi',
    '/EFI/almalinux/shim.efi',
    '/EFI/almalinux/grubx64.efi',
    '/EFI/debian/grubx64.efi',
    '/EFI/ubuntu/grubx64.efi',
    '/EFI/sles/grubx64.efi',
    '/EFI/opensuse/grubx64.efi',

'/EFI/Microsoft/boot/bootmgfw.efi'
  ]
-%>
```

BIOS UEFI e oVirt/VMWare

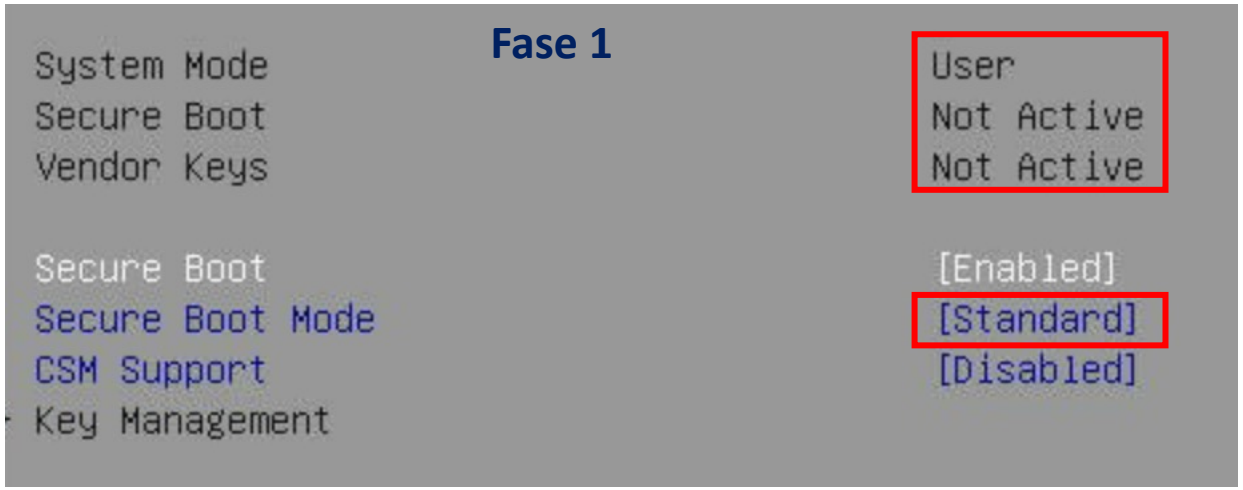
- Nei sistemi di virtualizzazione è necessario utilizzare il firmare adatto
  - oVirt/Openstack/KVM → Q35 Firmware with UEFI → Tianocore
  - VMWare → Firmware EFI



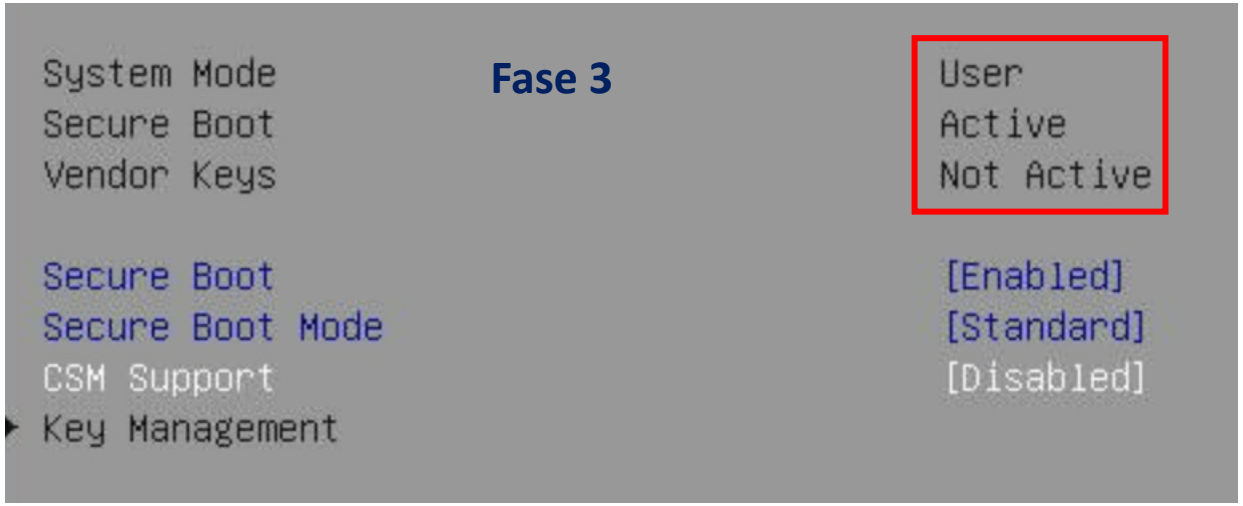
Secure Boot

- Modalità Standard
  - Uso di chiavi standard di verifica
- Setup richiede più fasi
  1. Abilitazione Secure Boot in modalità standard
  2. Fare installazione
  3. Al reboot secure boot è abilitato
- Al termine della procedura il boot via PXE non è più disponibile

Fase 1

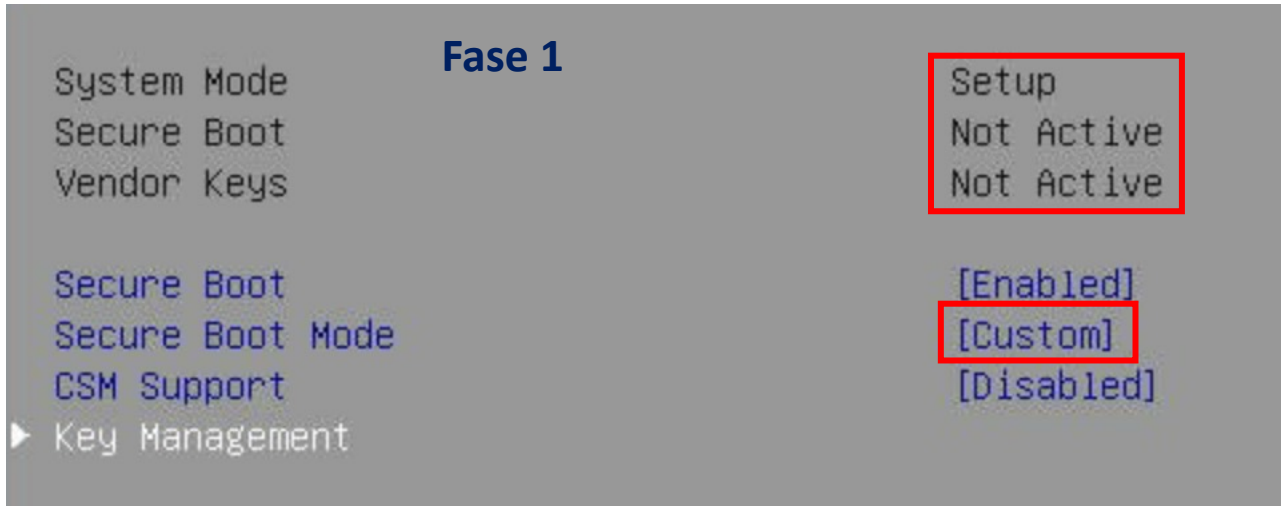


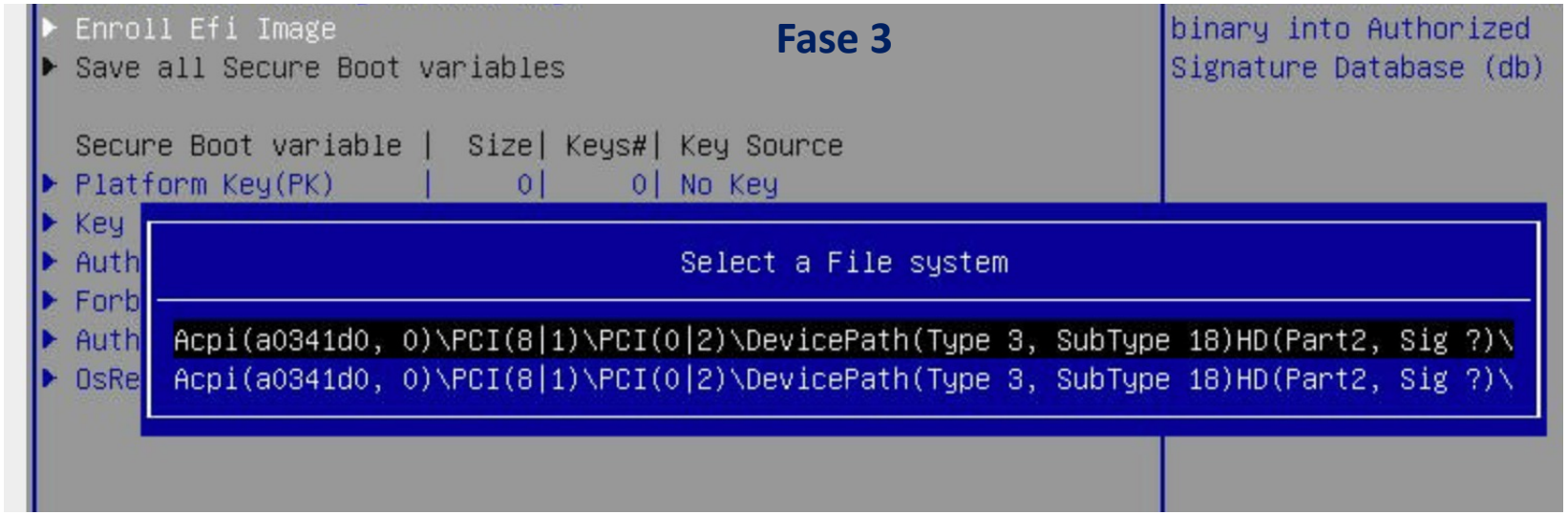
Fase 3



Secure Boot

- Modalità Custom
- Setup richiede più fasi
  1. Abilitazione Secure Boot in modalità custom
  2. Fare installazione
  3. Al reboot va importata la chiave di firma del bootloader del SO manualmente.
- Al termine della procedura il boot via PXE non è più disponibile





Conclusioni

- **Sempre più hardware** «particolare» richiede **UEFI**
- **Abbiamo dovuto imparare** come **fare installazioni unattended con Foreman** per macchine con UEFI
- Abbiamo deciso che **le nuove installazioni** che faremo nel nostro gruppo verranno **fatte con l'utilizzo di UEFI**
  - **Ultima Gara CPU** è stata installata con **UEFI**
- Visto che è il **nuovo standard** abbiamo deciso che anche le **VM su oVirt e VMWare dovranno essere create con firmware UEFI**.
- Faremo ancora qualche test con **SECURE BOOT** prima di usarlo in produzione
- Abbiamo in mente di **studiare** anche **HTTP BOOT**, possibilmente via HTTPS.

Domande


thank you
danke
謝謝
ngiyabonga
рахмат
teşekkür ederim
спасибо
Баярлалаа
gracias
merci
obrigado
dziękuję
bedankt
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
dank je
tapadh leat
hvala
mauruuru
sagolun
grazie
arigatô
takk
dakujem
merci
мерси
감사합니다
ευχαριστώ
faafetai lava
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
mamnun
дякую
djiere dieuf
tau
enkosi
bayarlalaa
gràcie
köszönöm
dankon
akun
aciù
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
dèkuji
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
ありがとう
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
merci
তোমাকে ধন্যবাদ
xièxie
trugarez
nanni
nandii
dhanyavad
kiitos
dankie
welalin
tack
barka
mersi
kia ora
spas