

Przygotowanie wzorcowego obrazu systemu typu Unix/Linux do wdrożenia w chmurze obliczeniowej

Przygotowanie środowiska

Należy przygotować wybrane obrazy dysków instalacyjnych typu iso, za pomocą których zostaną zainstalowane odpowiednie wersje systemów operacyjnych dostosowanych do wymagań wdrożenia w chmurze obliczeniowej.

Sam proces instalacji i konfiguracji może odbywać się na dowolnej maszynie z zainstalowanym systemem Linux, nie musi to być środowisko chmury. Po zakończeniu prac związanych z przygotowaniem właściwych obrazów, pliki zostaną przesłane do środowiska chmurowego, jako gotowe wzorcowe dyski instalacyjne lub uruchomieniowe.

Przygotowanie narzędzi

Należy zainstalować następujące programy: virtinst, virt-viewer, oraz opcjonalnie virt-manager:

```
apt-get install virtinst virt-viewer
```

Jeśli będziemy potrzebowali utworzyć nowy dysk z zainstalowanym systemem, na bazie gotowych obrazów iso, należy zainstalować następujące pakiety (w przypadku serwera będącego częścią chmury obliczeniowej instalacja nie jest to wymagane)

```
apt-get install qemu-utils
```

Instalacja i konfiguracja prostej kompilacji systemu na bazie dystrybucji Cirros

W pierwszym kroku należy pobrać obraz systemu z adresu:

- <https://download.cirros-cloud.net/0.6.3/>

i wybieramy dysk:

- `cirros-0.6.3-x86_64-disk.img`

Do zadań związanych z instalacją, konfiguracją i zarządzaniem środowiskiem maszyn wirtualnych, korzystamy z następujących narzędzi:

- `virt-install`
- `virsh`
- `qemu-img`

Program virt-install służy do wirtualizacji, za jego pomocą możemy przygotować działającą maszynę wirtualną, oraz wygenerować obraz zainstalowanego systemu. Program korzysta z mechanizmów wirtualizacji zależnych od możliwości systemu operacyjnego, może to być KVM, Qemu, lub inny mechanizm wirtualizacji.

Do zadań związanych z zarządzaniem, monitorowaniem parametrów wirtualizacji wykorzystuje się polecenie virsh.

Przykładowe polecenie związane z zarządzaniem instancjami maszyn wirtualnych:

- `virsh list --all`, wyświetla listę wszystkich instancji maszyn wirtualnych,
- `virsh destroy <nazwa_maszyny>`, zatrzymuje aktywną maszynę wirtualną,
- `virsh undefine <nazwa_maszyny>`, zwalnia nazwę skojarzoną z maszyną

Wykonanie samego polecenia `virsh` powoduje przejście do trybu interaktywnego menadżera, mamy wówczas możliwość bardzo szczegółowo zarządzać parametrami maszyny wirtualnej.

Wybrane polecenia wewnętrzne programu `virsh` związane z siecią i najczęściej wykorzystywane przy konfiguracji maszyn wirtualnych:

- | | |
|--------------------------------|--|
| • <code>net-autostart</code> | automatycznie uruchamia sieć |
| • <code>net-create</code> | utworzy sieć z pliku XML |
| • <code>net-define</code> | definiuje nieaktywną wirtualną sieć lub modyfikuje istniejącą, która jest zdefiniowana w pliku XML |
| • <code>net-destroy</code> | usuwa (zatrzymuje) sieć |
| • <code>net-dhcp-leases</code> | wyświetla informacje o przydzielonych ustawieniach |
| • <code>net-dumpxml</code> | informacje o sieci w XML |
| • <code>net-edit</code> | modyfikuje konfigurację XML dla sieci |
| • <code>net-event</code> | wyświetla informacje o zdarzeniach |
| • <code>net-info</code> | informacje o sieci |
| • <code>net-list</code> | lista sieci |
| • <code>net-name</code> | konwertuje identyfikator lub UUID sieci do nazwy sieci |
| • <code>net-start</code> | uruchamia (wcześniej określona) nieaktywna sieć |
| • <code>net-undefine</code> | unieważnia sieć |
| • <code>net-update</code> | aktualizuje konfigurację sieci |
| • <code>net-uuid</code> | konwertuje nazwę sieci na UUID |

- `iface-begin` tworzy snapshot aktualnych ustawien interfesu, który może zostać później zatwierdzony, wycofany, lub przywrócony,
- `iface-bridge` tworzy most między urządzeniem a istniejącą siecią,
- `iface-commit` zatwierdza ustawienia
- `iface-define` definiuje nieaktywny interfejs na podstawie pliku XML
- `iface-destroy` niszczy fizyczny interfejs gospodarza (wylacza go/„if-down”)
- `iface-dumpxml` informacje o interfejsie w XML
- `iface-edit` modyfikuje konfigurację XML dla fizycznego interfejsu gospodarza
- `iface-list` lista fizycznych interfejsów gospodarza
- `iface-mac` konwertuje nazwę interfejsu na jego adres MAC
- `iface-name` konwertuje adres MAC interfejsu na jego nazwę
- `iface-rollback` przywraca poprzednio zapisaną konfigurację stworzoną za pomocą `iface-begin`
- `iface-start` uruchamia fizyczny interfejs gospodarza (włącza go/„if-up”)
- `iface-unbridge` usuwa most po zwolnieniu zasobów przez podległe interfejsy
- `iface-undefine` usuwa określenie fizycznego interfejsu gospodarza (usuwa je z konfiguracji)

Instalacja / konfiguracja systemu

A rozpocząć proces instalacji, należy wykonać polecenie (zalecane jest wykorzystanie klienta SSH z obsługą X11):

```
virt-install --name=my_cirros --ram=256 --vcpus=1 \
--disk ./cirros-0.6.3-x86_64-disk.img,format=qcow2 \
--import --network=network=default
```

w nowszych wersjach `virt-install` należy użyć składni:

```
virt-install --name=my_cirros --ram=256 --vcpus=1 \
--disk ./cirros-0.6.3-x86_64-disk.img,format=qcow2 \
--import --network=network=default --osinfo=cirros0.5.2
```

Jeśli podczas uruchamiania instalatora pojawi się komunikat:

```
Cannot access storage file '.../cirros-0.6.3-x86_64-disk.img' (as  
uid:64055, gid:994): Permission denied
```

czyli błąd związany z brakiem uprawnień użytkownika do dostępu do pliku `cirros-0.6.3-x86_64-disk.img`, to należy wykonać polecenie:

```
setfacl -m u: libvirt-qemu:rwx ../
```

lub w nowszych wersjach systemu Linux:

```
setfacl -m libvirt-qemu:rwx ../
```

znaczniki przełączników:

- `name` - nazwa maszyny,
- `ram` - ilość pamięci przydzielonej maszynie,
- `vcpus` - ilość rdzeni procesora przydzielonych maszynie,
- `disk` - ścieżka do dysku z systemem lub dysku do instalacji systemu,
- `import` - określenie dysku jako istniejącego, z którego zostanie uruchomiony system,
- `network` - konfiguracja sieci

Jeśli domyślne ustawienia sieci nie działają poprawnie, możemy bardziej szczegółowo skonfigurować połączenie sieciowe. Ma to duże znaczenie w przypadku systemów typu "net-install", które do poprawnego zainstalowania systemu wymagają dostępu do sieci internet.

Polecenia (`virsh`) które mogą być przydatne w diagnostyce konfiguracji sieci:

- `net-list`
- `net-info`,
- `net-dumpxml`
- `net-edit`
- `iface-list`
- `iface-dumpxml`
- `iface-edit`
- `iface-define`
- `iface-begin`
- `iface-start`
- `iface-commit`

W przykładowej konfiguracji, skorzystamy z faktu, iż polecenia wydajmy na serwerze który jest związany z chmurą.

Interfejs który chcemy podłączyć do maszyny wirtualnej to `enp0s3`. Interfejs `enp0s3` jest podłączony do sieci internet za pomocą NAT, oraz zawiera kilka wirtualnych mostków typu `virbr0`.

Na początek należy przygotować plik XML z konfiguracją sieci:

Plik: siec.xml

```
<network>
  <name>siec</name>
  <forward mode="bridge">
    <interface dev="enp0s3"/>
  </forward>
</network>
```

Wykorzystamy tutaj własność bezpośredniego mapowania interfejsów sieciowych hosta na wirtualny interfejs gościa. Mechanizm nazywa się "macvtap".

W przypadku wirtualizacji w oparciu o KVM/QEMU mamy do dyspozycji następujące mechanizmy mapowania:

- macvtap, pozwala na bezpośrednie połączenie między interfejsami hosta i gościa,
- vepa, przekierowanie wszystkich wywołań gościa na wskazany zewnętrzny adres mostu,
- bridge, obywie sieci hosta i gościa muszą być typu most,
- private, wszystkie pakiety są wysyłane do zewnętrznego mostu, dane do gościa są dostarczane tylko wtedy, gdy trafią do zewnętrznego routera lub bramy i mogą być odesłane,
- passthrough, wszystkie pakiety są przekierowane na wybrany interfejs, dodatkowo można korzystać z mechanizmów nasłuchiwania, zmiany adresów MAC.

Więcej szczegółów można znaleźć pod adresem: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/virtualization_deployment_and_administration_guide/sect-managing_guest_virtual_machines_with_virsh-interface_commands

Przed uruchomieniem konfiguracji należy sprawdzić, czy jądrę systemu linux posiada załadowany moduł obsługi połączeń tego typu:

- `lsmod | grep macvlan`

Jeśli nie został wyświetlony działający moduł, należy go załadować poleceniem:

- `modprobe macvlan`

Następnie konfigurujemy wirtualną sieć:

- `virsh net-define siec.xml`
- `virsh net-autostart siec`
- `virsh net-start siec`

Jeśli wszystko przebiegło poprawnie, możemy sprawdzić, czy nasza sieć działa zgodnie z założeniami:

- `virsh net-list`
- `virsh net-dumpxml siec`

Możemy teraz uruchomić ponownie maszynę, z nową konfiguracją sieci:

```
virt-install --name=cirros --ram=256 --vcpus=1 \
--disk ./cirros-0.6.3-x86_64-disk.img,format=qcow2 \
--import -network=network:siec,model=virtio
```

w nowszych wersjach `virt-install` należy użyć składni:

```
virt-install --name=my_cirros --ram=256 --vcpus=1 \
--disk ./cirros-0.6.3-x86_64-disk.img,format=qcow2 \
--import --network=network:siec,model=virtio --osinfo=cirros0.5.2
```

Wprowadzimy pewne modyfikacje w uruchomionej maszynie, aby się przekonać, że przygotowany na podstawie tej instancji obrazu dedykowany do wdrażania w chmurze obliczeniowej zachowa wprowadzone zmiany. Do celów testowych wprowadzimy zmianę hasła domyślnego użytkownika 'cirros' z wartości domyślnej na nową wartość 'hasło'.

Po zalogowaniu do maszyny wydajemy polecenia:

```
$ sudo su
```

```
$ passwd cirros
```

```
New password: hasło
```

```
Retype password: hasło
```

Aby hasło było widoczne w oknie powitalnym, edytujemy plik `/etc/issue` i wpisujemy zawartość:

```
login as 'cirros' user. default password: 'hasło' . use 'sudo' for
root.
```

Jeśli maszyna jest gotowa do wdrożenia, zamykamy system (z poziomu administratora):

```
poweroff
```

a na hoście wykonujemy polecenia:

```
apt-get install guestfs-tools
```

```
virt-sysprep -d my_cirros
```

Przygotowany obraz możemy przesłać do chmury, celem sprawdzenia działania dedykowanej wersji obrazu.

Instalacja i konfiguracja dystrybucji Ubuntu na bazie obrazu ISO

Na początku należy przygotować dysk, na którym zostanie zainstalowany system operacyjny,

```
qemu-img create -f qcow2 server.qcow2 8G
```

i przystępujemy do instalacji systemu:

```
virt-install --name base_server --ram 1024 \  
--cdrom=./Ubuntu_18_04_AMD64_mini.iso \  
--disk path=./server.qcow2,format=qcow2 \  
--network=network:siec,model=virtio
```

Po uruchomieniu nowego systemu należy zainstalować z poziomu administratora następujący pakiet (w maszynie wirtualnej) :

```
virt-install --name base_server --ram 1024 \  
--disk path=./server.qcow2,format=qcow2 \  
--import \  
--network=network:siec,model=virtio --osinfo=ubuntu18.04
```

- `apt-get install cloud-init`

Aby dostosować sposób uruchamiania systemu w chmurze do własnych potrzeb, należy dokonać modyfikacji w pliku `/etc/cloud/cloud.cfg` pliku obrazu maszyny wirtualnej. Warto zwrócić uwagę na pozycję `datasource`, można tutaj ustawić typ zgodności z popularnymi obrazami systemów, np. zezwolić na zgodność z maszynami typu EC2. Openstack może korzystać z metadanych w standardzie EC2, powszechnie stosowanych w chmurze publicznej Amazon AWS. Zapewnia kompatybilność naszych obrazów z obrazami chmury AWS, może być przydatne w przypadku migracji z chmury prywatnej do publicznej lub odwrotnie.

Warto również sprawdzić, czy użytkownik który jest zarejestrowany w systemie, jest zgodny z tym, który będzie wykorzystywany do połączeń szyfrowanych w oparciu o mechanizm SSH.

Cloud-init utworzy automatycznie użytkownika, który jest przypisany do pary kluczy SSH. Ostatnim krokiem, związanym z przygotowaniem dedykowanego obrazu systemu dla chmury obliczeniowej, jest zainstalowanie wymaganym usług, np. mogą to być serwisy typu:

- serwer WWW,
- serwer pocztowy,
- serwer plików,
- serwer bazy danych,
- interfejs webowy do zarządzania wybraną aplikacją / serwisem

Szczegółowa lista parametrów, jakie można ustawić za pomocą cloud-init znajduje się w dokumentacji pod adresem:

<https://cloudinit.readthedocs.io/en/latest/>

Jeśli maszyna jest gotowa do wdrożenia, zamykamy system (z poziomu administratora):

- `shutdown -h now`

a na hoście wykonujemy polecenie:

- `virt-sysprep -d base_server`

Jeśli ostatnie polecenie nie zostanie wykonane poprawnie, należy zainstalować następujący pakiet:

- `apt-get install libguestfs-tools`