univention
be open.

Univention Corporate Server

Extended installation documentation

univention
be open.

Table of Contents

Chapter 1. Using a UCS appliance

In addition to the traditional installation, there is also the possibility of providing UCS via an appliance image. These appliance images can be used both for simple commissioning in a virtualization solution such as VMware and for providing a cloud instance.

Appliances can be created with minimal effort. This is described in Chapter 2.

Whilst some of the settings can be preconfigured globally in the image, it is still necessary for the end user to make final adjustments to the configuration, e.g., to set the computer name or the domain used. For this reason, a basic system is installed for the appliance image and a component set up, which then allows the end user to finalise the configuration. Alternatively, the configuration can also be performed automatically without user interaction. This is described in Section 2.3.

The interactive configuration can be performed in two ways:

- A graphic interface starts on the system, in which the web browser Firefox is started in full-screen mode and automatically accesses the configuration URL. This option is particularly suitable for images in virtualization solutions.
- The configuration can also be performed directly via an external web browser. In this case, the system's IP address must be known to the user (e.g., if it has been notified to him in advance in the scope of the provision of a cloud image).

In the scope of the initial configuration, the user can change the following settings in the default setting:

- Selection of the language, time zone and keyboard layout
- Configuration of the network settings
- Setup of a new UCS domain or joining a UCS or Microsoft Active Directory domain
- Software selection of UCS key components. The user can install software from other vendors at a later point in time via the Univention App Center.

Chapter 2. Creating a UCS appliance/ cloud image

2.1. Introduction

Feedback

This article describes how to set up an appliance based on UCS 4.0. This type of appliance can also be used to provide preconfigured instances as a cloud service provider. The creation of images for typical virtualization solutions is another possible application scenario, see Section 2.2.3.

2.2. Performing the basic installation

Feedback

The basic installation is performed using the standard UCS installer. Further information on the individual options can be found in the UCS manual. The installation should be performed in a virtualization solution. In this example, the installation is performed in UVMM. A qcow2 image should be selected for the hard drive for the virtual machine. Qcow2 images can be converted to different virtualization formats such as VirtualBox or VMware using a tool provided by Univention, see Section 2.2.3.

The following settings are configured for the basic image:

- The installation language can be selected as required. The locale of the system is set based on the selected language. If you want to be able to use the appliance in more than one language, you can add another locale at a later point in time.
- A preselection is made for the time zone which is then adapted subsequently by the users of the appliance.
- The keyboard layout is only relevant for local logins; it is not important for the web-based configuration.
- A configuration via DHCP is the most practical presetting for appliance images. The Univention Installer attempts to perform a DHCP request in the scope of the network configuration. The network configuration is only performed via DHCP if this is successful, i.e., an IP address must be assigned to the appliance for the duration of the setup. This can be done with an *IP managed client* object in the Univention Management Console.
- In the next step, the initial password is set for the root user. This root password is changed by the end user during the commissioning of the appliance image.
- The partitioning can be performed as required, e.g., by using an LVM. For an image that will be used in a cloud setup, a single root partition should be used. This allows growing the root partition based on the selected instance disk size.

Once the basic installation is complete, a dialogue is shown in which you can select whether to create a new UCS domain or join an existing domain. To create the appliance, **Control+Q** must be pressed at this point to interrupt the process. The installation continues for a short period of time, during which the **Starting Univention System Setup** message appears and the systems then restarts.

The installation of the basic image is now complete. Following a reboot, the user of the appliance is shown the dialogue for adjusting the configuration, see Chapter 1.

In most cases, the appliance needs to be preconfigured with a certain selection of software. The installation is usually performed via the Univention App Center, which, however, is not yet available at this point in time. The installation is thus performed via the command line. UCS standard components can be installed using the corresponding package names, e.g.

```
univention-install univention-printserver
```

Packages from the Univention App Center are installed with the command `univention-add-app` once a valid license is available. The ID of an application can be retrieved with the command `univention-add-app --list`:

```
univention-add-app -l APPID
```

The system now needs to be shut down cleanly without filesystems still being mounted.

The qcow2 image (i.e., the hard drive of the virtual machine) is now copied. If the *default* storage pool of UVMM was used, the image is stored in the directory `/var/lib/libvirt/images`.

Additional steps are required if the image is to be used in Amazon EC2 (see Section 2.2.1), OpenStack (see Section 2.2.2) or as a VMware / VirtualBox appliance (see Section 2.2.3).

2.2.1. Providing an image for Amazon EC2

Feedback

The following adjustments need to be made for an image that is to be used in Amazon EC2.

Amazon EC2 uses a PyGrub version which can only read the data format from Grub 1 (`menu.lst`), whilst UCS uses Grub 2 as the bootloader. The following Univention Configuration Registry variables can be used to generate the Grub configuration in this format additionally. The bootloader configuration is also adapted:

```
append="$(ucr get grub/append)"
ucr set grub/append="$(echo "$append" |
 sed -e 's|/dev/sda|/dev/xvda|g;s|splash|nosplash|')"
ucr set grub/root=/dev/xvda1
ucr set grub/grub1root="(hd0)"
update-initramfs -uk all
update-grub
univention-grub-generate-menu-lst
```

The initial login to the EC2 instance is performed via a SSH host key. To prevent SSH logins from occurring with the default root password of the standard image during commissioning of the instance, the initial root password is removed. The following Univention Configuration Registry variable configures this start mode:

```
usermod -p \* root
ucr set server/amazon=true
```

The name server should be set; in this example to `OpenDNS`. Additionally, the timeout when waiting for a DHCP request answer is lowered.

```
ucr set nameserver1=208.67.222.222 dns/forwarder1=208.67.222.222
ucr unset nameserver2 nameserver3
ucr unset dns/forwarder2 dns/forwarder3
ucr set interfaces/eth0/type=dhcp dhclient/options/timeout=12
```

2.2.2. Providing an image for OpenStack

Feedback

The provisioning for OpenStack images occurs via Cloud-Init (see Section 2.3.2). Cloud-Init is a standardised solution for configuration of an image. Cloud-Init checks a range of data sources for an existing configuration. The ***univention-cloud-init*** package must be installed to prepare an image for provisioning via Cloud-Init:

```
univention-install univention-cloud-init
```

The local Firefox session should not be started when running as an OpenStack instance.

```
ucr set system/setup/boot/start=false
```

The initial login to the OpenStack instance is performed via a SSH host key. To prevent SSH logins from occurring with the default root password of the standard image during commissioning of the instance, the initial root password is removed.

```
usermod -p \* root
```

2.2.3. Providing an image for VMware/VirtualBox

Feedback

Virtualization images for VirtualBox, VMware Player and VMware ESX can also be created on the basis of the qcow2 images above. To this end, Univention offers a tool, which can be installed via the ***generate-appliance*** package (the integration in UCS 4.0 can be followed via Bug 37137).

The `generate_appliance` tool must be started and the qcow2 image selected with the parameter *-s*:

```
generate_appliance -s appliance.qcow2
```

The virtual machine is assigned one CPU and a gigabyte of RAM as standard. If the appliance has a higher storage or CPU power requirement, the parameter *-m* can be used to specify a different quantity of RAM in megabytes and *-c* can be used to assign a different number of CPUs. The parameters *--vendor* and *--product* can be used to specify a vendor and product name.

In the default setting, three different virtualization images are generated from the qcow2 image. The generation for a type can be suppressed using the respectively given option:

- Zipped VMware compatible images (e.g. for VMware Player), can be suppressed with `--no-vmware`
- VirtualBox OVA image, can be suppressed with `--no-ova-virtualbox`
- VMware ESX OVA image, can be suppressed with `--no-ova-esxi`

2.3. Automatic configuration of an appliance

Feedback

Instead of an interactive configuration of the appliance by the user, it can also be performed automatically. The automatic configuration can either be performed via cloud-init (a general tool for the provision of cloud images) or a Univention appliance mode profile file.

2.3.1. Automatic configuration with a UCS appliance mode profile file

Feedback

Automatic configuration with the UCS appliance mode requires creating a profile file `/var/cache/univention-system-setup/profile`. Example configuration:

```
hostname="ucs"
domainname="testdom.local"
windows/domain="TESTDOM"
ldap/base="dc=testdom,dc=local"
root_password="univention"

locale/default="de_DE.UTF-8:UTF-8"
components="univention-s4-connector:univention-samba4 univention-nagios-
server"
packages_install="univention-s4-connector univention-samba4 univention-
nagios-server"
packages_remove=""

server/role="domaincontroller_master"
```

```
interfaces/eth0/type=""
interfaces/eth0/address="10.201.101.2"
interfaces/eth0/netmask="255.0.0.0"
interfaces/eth0/network="10.0.0.0"
interfaces/eth0/broadcast="10.255.255.255"
dns/forwarder1="10.201.74.2"
gateway="10.201.0.1"
```

If `interfaces/eth0/type` is set to `dynamic`, DHCP is used for the network configuration.

Then the `/usr/lib/univention-system-setup/scripts/setup-join.sh` tool needs to be run once. Then Apache and the UMC server need to be restarted:

```
invoke-rc.d apache2 restart
invoke-rc.d univention-management-console-server restart
```

2.3.2. Automatic configuration of an appliance with Cloud-Init

Cloud-Init works on a configuration file in the cloud configuration format. The configuration file is provided by the respective cloud service; the type of provision differs from cloud solution to cloud solution. It is currently only possible to provide a master domain controller.

The configuration file may be adapted for different scenarios. To setup a domain, the `ucs_setup` section is required. Note that the supplied `ldap_base` is used in other configuration sections as well.

The following includes an example file with which a master domain controller can be provided. In addition, several files are generated on the system: the UCS license to be installed and a file with the apps to be installed from the Univention App Center. The license in this example is the default *free for personal use license*. More information about requesting a proper license can be found in Section 2.3.3.

Two example hook scripts are generated which are called after setup is finished: One calls `wget` for a given URL, which could be used to signal an external service that the provisioning of the instance is done. The other is an example `udm` call to register an OpenStack connection in UVMM.

```
#cloud-config
#
ucs_setup:
  hostname: myucsmaster
  domainname: ucs.local
  windowsdomain: UCS
  ldap_base: dc=ucs,dc=local
  rootpassword: univention
  defaultlocale: de_DE.UTF-8:UTF-8
  components:
  packages_install: univention-virtual-machine-manager-daemon
  packages_remove:
write_files:
-   content: |
      dn: cn=admin,cn=license,cn=univention,dc=ucs,dc=local
      objectClass: top
      objectClass: univentionLicense
      objectClass: univentionObject
      univentionObjectType: settings/license
      univentionLicenseEndDate: unlimited
      univentionLicenseModule: admin
      cn: admin
      univentionLicenseBaseDN: Free for personal use edition
```

```
      univentionLicenseUsers: 50
      univentionLicenseServers: unlimited
      univentionLicenseManagedClients: 50
      univentionLicenseCorporateClients: unlimited
      univentionLicenseVirtualDesktopUsers: unlimited
      univentionLicenseVirtualDesktopClients: unlimited
      univentionLicenseSupport: 0
      univentionLicensePremiumSupport: 0
      univentionLicenseVersion: 2
      univentionLicenseType: UCS
      univentionLicenseSignature: FPkmAy7icL0N/2w15fREr/
l4rD9r6xaLi3C7dLRVQbDX5zdcQ+
       0BSPfHFT8pdAaMpZqXkp13rQK4N2PzHHPy0HwyyaeMTckxJyaIQd/B/M4t+vbIxnz
+2YNoDVqskws
       me/
V8qxgoMsMFT47LUDp7serXZoCETutwcwwqrycy0RdahezzyAA93e3bBaH1y5zK1b4ElAWW933d
       fctWTR+/LdQglhKbRlHBN8fbcHLmXBtu7yMV3J7AGl
+wTo4+KzASUWdTmrfUY5dcU77spyCMJEGL6
       ZxsFs8FhDrYajyAn2s0rAcxJK9JKFFjBuuiF+8MB2dyypkJilSiTRQDhEYorQ==
    owner: root:root
    path: /var/cache/univention-system-setup/license
    permissions: '0400'
-   content: |
      simplesamlphp
      adconnector
    owner: root:root
    path: /var/cache/univention-system-setup/installapps
    permissions: '0400'
-   content: |
      #!/bin/sh
      wget http://myURL/page?myparam=myValue
    owner: root:root
    path: /usr/lib/univention-system-setup/appliance-hooks.d/90_wget_url
    permissions: '0755'
-   content: |
      #!/bin/sh
      udm uvmm/cloudconnection create --ignore_exists \
      --position="cn=CloudConnection,cn=Virtual Machine
 Manager,dc=ucs,dc=local" \
      --set name="OpenStack" \
      --set type="cn=OpenStack,cn=CloudType,cn=Virtual Machine
 Manager,dc=ucs,dc=local" \
      --set includeUCSimages=0 \
      --append parameter="username demouser" \
      --append parameter="password password" \
      --append parameter="cloudtype OpenStack" \
      --append parameter="auth_url http://192.168.0.1:5000" \
      --append parameter="auth_version 2.0_password" \
      --append parameter="service_type compute" \
      --append parameter="service_name nova" \
      --append parameter="tenant demotenant" \
      --append parameter="service_region regionOne"
      invoke-rc.d univention-virtual-machine-manager-daemon restart
    owner: root:root
```

```
    path: /usr/lib/univention-system-setup/appliance-
hooks.d/80_add_uvmm_connection
    permissions: '0755'
```

The file with the apps to be installed contains a list of IDs of applications from the Univention App Center, see Section 2.2. The list in the example above installs the Univention AD Connector and the SAML integration on the provided master domain controller

2.3.3. License management in cloud instances

Feedback

In the default installation, a UCS installation has a so-called *free for personal use license* (up to fifty users). A license from Univention is required for the productive use of a UCS installation. For standard installations it is sent to the user by e-mail and then set up in the Univention Management Console.

Cloud service providers have the possibility of retrieving UCS licenses via an API, i.e., if a new instance is to be created for a customer, the license can be retrieved via the API and then installed in the provided instance directly.

Access to the license server requires a user name and a password. These can be requested from Univention at sales@univention.de. In this document, `https://license.univention.de/shop/example/` is used as an example URL for the license server.

2.3.3.1. API for retrieving UCS licenses

Feedback

The licenses are retrieved via HTTPS from the Univention license server `license.univention.de`. The retrieval can be performed completely with `wget`.

Firstly, a session with the license server must be opened, in this case with the user name `univention` and the password `secret` as an example. It is also possible to request more than one license in one session.

```
wget --keep-session-cookies --save-cookies cookie.db --load-cookies \
 cookie.db --post-data='username=univention&password=secret' \
 https://license.univention.de/shop/example/
```

A license can also be ordered with a POST request via `wget`. Please note that special characters such as blank spaces must be escaped in URL-encoded syntax, see `http://en.wikipedia.org/wiki/Percent-encoding` for details.

```
wget --keep-session-cookies --save-cookies cookie.db --load-cookies
 cookie.db \
--post-data='kundeEmail=customer@example&'\
'kundeUnternehmen=New%20Customern&'\
'EndDate=27.11.2015&'\
'BaseDN=dc%3Ddrei%2Cdc%3Dzwei%2Cdc%3Dtest&'\
'Servers=0&'\
'Support=0&'\
'PremiumSupport=0&'\
'Users=100&'\
'ManagedClients=0&'\
'CorporateClients=0&'\
'VirtualDesktopUsers=0&'\
'VirtualDesktopClients=0&'\
'Type=UCS' \
https://license.univention.de/shop/example/order
```

If the order is successful, the return code 202 is returned. The HTML data includes the tag `orderid`, which identifies the order number of a successful order:

```
...
<span id="orderid">21</span>
...
```

If the order fails, a return code 4xx is returned and the `details` tag includes additional information, e.g.:

```
...
<span id="details">Not a valid date: u'27.11.201'</span>
...
```

Should it not be possible to process an order due to a server error, 5xx is output as the return code. The order can then be repeated at a later point in time.

Following ordering of a license, it takes a few seconds before the license is generated. It can then be retrieved in LDIF format using the order number. If the request above returns e.g. the order number 465, the file name is thus `465.ldif`. The request specified below waits for the availability of the license for up to sixty seconds:

```
wget --keep-session-cookies --save-cookies cookie.db --load-cookies
 cookie.db \
https://license.univention.de/shop/example/orders/465.ldif
```