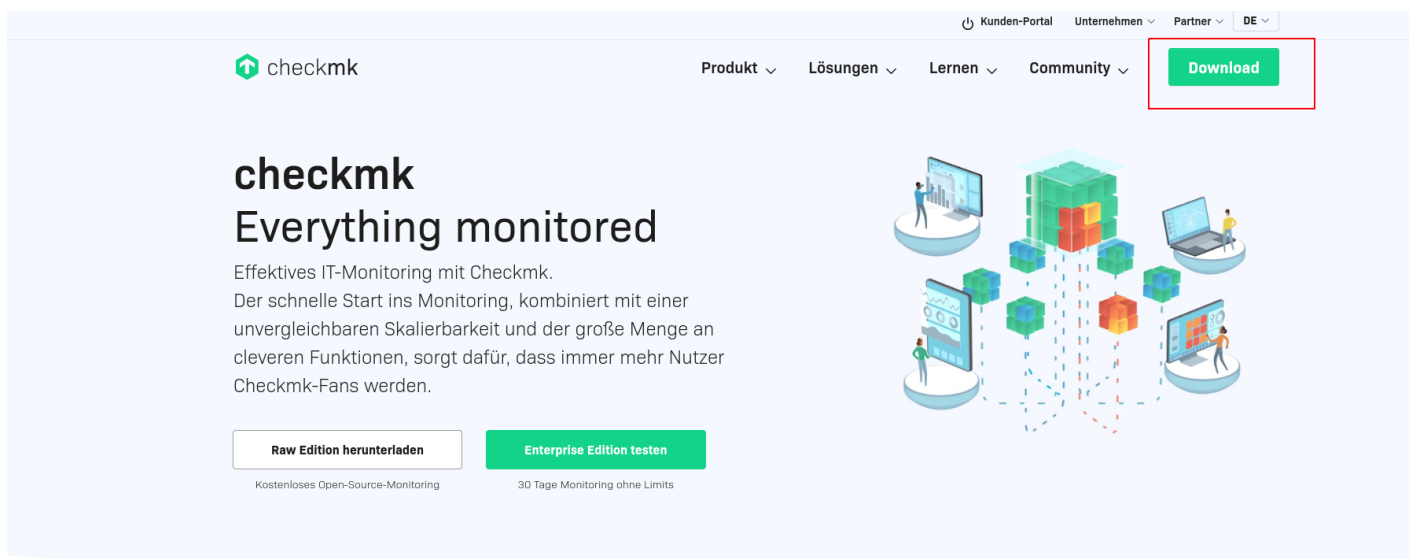


Installation auf Redhat / CentOS 8

1. Voraussetzung ist ein Debian 11 mit ssh Zugang via Schlüsseldatei.

Auf der Website <https://checkmk.com/de>

Und oben rechts auf Downloads.



Nun Checkmk für Linux auswählen -> Free / Enterprise -> Version 2.1 bzw die Aktuellste -> Red Hat /Centos -> red Hat / Alma Linux -> geschäftlich

Emailadresse (Kann auch ne Fake Email sein, denn die Installationsanleitung kommt danach eingeblendet)

eintragen und auf Download & install drücken

Checkmk für Linux	Checkmk für Docker	Checkmk Appliance
----------------------	-----------------------	----------------------

1. Edition wählen

Raw	Free / Enterprise Trial	Enterprise	Managed Services
-----	-------------------------	------------	------------------

2. Plattform wählen

VERSION	PLATTFORM	OS-VERSION	ART DER NUTZUNG	FIRMEN-E-MAIL
2.1.0p20	Ubuntu	Red Hat / AlmaLinux	geschäftlich	<input type="text" value="E-Mail-Adresse eingeben"/> <input type="button" value="Download & Install"/>
2.0.0p33	Debian	Red Hat / CentOS 8.x	persönlich	
1.6.0p30	SLES	Red Hat / CentOS 7.x		
	RedHat/CentOS			

Jetzt werden die Befehle zur Installation angezeigt. Ich habe diese aber zum kopieren im nächsten Schritt angefügt

3. Checkmk installieren

3.1 Einrichtung der Paketquellen, von SELinux und der Firewall

Checkmk benötigt etliche Softwarepakete von Ihrer Linux-Distribution. Software aus Drittquellen wird nicht benötigt. Damit Sie alle benötigten Pakete sauber nachinstallieren können, benötigen Sie eine korrekte Konfiguration der Softwarequellen, sowie von SELinux und der Firewall. Befolgen Sie dazu die [hier](#) beschriebenen Schritte.

3.2 Checkmk für Red Hat oder CentOS downloaden

Das passende Paket via *wget* herunterladen

[Changelog](#)

```
$ wget https://download.checkmk.com/checkmk/2.1.0p20/check-mk-free-2.1.0p20-el8-38.x86_64.rpm
```

copy



SHA-256 Datei Hash:

```
1cb92a6cebb7f0c5861358d3bed5ece56b92d240c0872ea9be24c059ff0779d9
```

copy

3.3 Installation des signierten Pakets

Alle Pakete werden mittels GnuPG signiert. Durch diese Signatur können Sie zum einen prüfen, ob das Paket auch wirklich von uns stammt und zum anderen, ob das Paket insgesamt vollständig ist. Befolgen Sie dazu die [hier](#) beschriebenen Schritte.

3.4 Das Checkmk Paket installieren

Nun installieren Sie das Paket mit all seinen Abhängigkeiten.

```
$ sudo rpm --install ./check-mk-free-2.1.0p20-el8-38.x86_64.rpm
```

copy

Die Debdatei per WGET downloaden. Zur Zeitpunkt der Erstellung des Artikels Version 2.1

```
wget https://download.checkmk.com/checkmk/2.1.0p20/check-mk-free-2.1.0p20-el8-38.x86_64.rpm
```

Nun das Paket installieren

```
rpm --install ./check-mk-free-2.1.0p20-el8-38.x86_64.rpm
```

Wenn wir einen Haufen dependencies Fehler bekommen. Sind die Abhängigkeiten noch nicht installiert

```
warning: ./check-mk-free-2.1.0p20-el9-38.x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID c4503261:  
NOKEY
```

error: Failed dependencies:

dialog is needed by check-mk-free-2.1.0p20-el9-38.x86_64

freeradius-utils is needed by check-mk-free-2.1.0p20-el9-38.x86_64

graphviz-gd is needed by check-mk-free-2.1.0p20-el9-38.x86_64

libgsf is needed by check-mk-free-2.1.0p20-el9-38.x86_64

perl-IO-Zlib is needed by check-mk-free-2.1.0p20-el9-38.x86_64

perl-Locale-Maketext-Simple is needed by check-mk-free-2.1.0p20-el9-38.x86_64

```
perl-Net-Ping is needed by check-mk-free-2.1.0p20-el9-38.x86_64
poppler-utils is needed by check-mk-free-2.1.0p20-el9-38.x86_64
rpm-build is needed by check-mk-free-2.1.0p20-el9-38.x86_64
rsync is needed by check-mk-free-2.1.0p20-el9-38.x86_64
time is needed by check-mk-free-2.1.0p20-el9-38.x86_64
uuid is needed by check-mk-free-2.1.0p20-el9-38.x86_64
```

Nun mit dem Paketmanger dnf installieren, der installiert die Abhängigkeiten gleich mit.

```
dnf install check-mk-free-2.1.0p20-el8-38.x86_64.rpm
```

Sollte da folgender Fehler kommen:

```
CentOS Linux 8 - AppStream                               157 B/s | 38 B   00:00
Error: Failed to download metadata for repo 'appstream': Cannot prepare internal mirrorlist: No URLs in mirrorlist
```

Cent OS 8, ist eigentlich End of Life. Nun die Repo austauschen um das Cent OS 8 zu aktualisieren.

```
sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
```

Nun den DNF Befehl nochmals ausführen und alle Abhängigkeiten werden mit installiert.
Ausgabe:

```
[root@catl02v406 ~]# dnf install check-mk-free-2.1.0p20-el8-38.x86_64.rpm
CentOS Linux 8 - AppStream                               11 MB/s | 8.4 MB   00:00
CentOS Linux 8 - BaseOS                                 15 MB/s | 4.6 MB   00:00
CentOS Linux 8 - Extras                                 124 kB/s | 10 kB   00:00
CentOS Linux 8 - PowerTools                             8.6 MB/s | 2.3 MB   00:00
ICINGA (stable release for epel)                       61 kB/s | 2.9 kB   00:00
Extra Packages for Enterprise Linux Modular 8 - x86_64  53 kB/s | 30 kB    00:00
Extra Packages for Enterprise Linux Modular 8 - x86_64  1.1 MB/s | 733 kB  00:00
Extra Packages for Enterprise Linux 8 - x86_64         45 kB/s | 24 kB    00:00
Extra Packages for Enterprise Linux 8 - x86_64
....

Upgraded:
elfutils-libelf-0.185-1.el8.x86_64      elfutils-libs-0.185-1.el8.x86_64  graphviz-2.40.1-43.el8.x86_64  ima-
evm-utils-1.3.2-12.el8.x86_64  openssl-1:1.1.1k-5.el8_5.x86_64  openssl-devel-1:1.1.1k-5.el8_5.x86_64
openssl-libs-1:1.1.1k-5.el8_5.x86_64  python3-rpm-4.14.3-19.el8.x86_64  rpm-4.14.3-
```

```
19.el8.x86_64 rpm-build-libs-4.14.3-19.el8.x86_64 rpm-libs-4.14.3-19.el8.x86_64 rpm-plugin-selinux-
4.14.3-19.el8.x86_64
rpm-plugin-systemd-inhibit-4.14.3-19.el8.x86_64
```

Installed:

```
bzip2-1.0.6-26.el8.x86_64 check-mk-free-2.1.0p20-el8-
38.x86_64 dialog-1.3-13.20171209.el8.x86_64 elfutils-0.185-1.el8.x86_64
freeradius-3.0.20-10.module_el8.5.0+1057+66764497.x86_64 freeradius-utils-3.0.20-
10.module_el8.5.0+1057+66764497.x86_64 gc-7.6.4-3.el8.x86_64 gdb-headless-8.2-
16.el8.x86_64
graphviz-gd-2.40.1-43.el8.x86_64 guile-5:2.0.14-7.el8.x86_64
libatomic_ops-7.6.2-3.el8.x86_64 libbabeltrace-1.5.4-3.el8.x86_64
libgsf-1.14.41-5.el8.x86_64 libipt-1.6.1-8.el8.x86_64 nspr-
4.32.0-1.el8_4.x86_64 nss-3.67.0-7.el8_5.x86_64
nss-softokn-3.67.0-7.el8_5.x86_64 nss-softokn-freebl-3.67.0-
7.el8_5.x86_64 nss-sysinit-3.67.0-7.el8_5.x86_64 nss-util-3.67.0-7.el8_5.x86_64
patch-2.7.6-11.el8.x86_64 perl-DBI-1.641-
3.module_el8.3.0+413+9be2aeb5.x86_64 perl-IO-Zlib-1:1.10-420.el8.noarch perl-Locale-
Maketext-1.28-396.el8.noarch
perl-Locale-Maketext-Simple-1:0.21-420.el8.noarch perl-Math-BigInt-1:1.9998.11-
7.el8.noarch perl-Math-Complex-1.59-420.el8.noarch perl-Net-Ping-2.55-420.el8.noarch
perl-Time-HiRes-4:1.9758-2.el8.x86_64 poppler-20.11.0-
3.el8_5.1.x86_64 poppler-data-0.4.9-1.el8.noarch poppler-utils-20.11.0-
3.el8_5.1.x86_64
rpm-build-4.14.3-19.el8.x86_64 rsync-3.1.3-12.el8.x86_64
time-1.9-3.el8.x86_64 tpm2-tss-2.3.2-4.el8.x86_64
uuid-1.6.2-43.el8.x86_64 xinetd-2:2.3.15-24.el8.x86_64
zstd-1.4.4-1.el8.x86_64
```

Complete!

Überprüfen ob korrekt installiert wurde

```
omd version
```

Die Ausgabe sollte so aussehen

```
OMD - Open Monitoring Distribution Version 2.1.0p16.cfe
```

Nun eine Checkmk instanz erstellen. Eine Intenz wäre zum Beispiel Kunde oder ein Projekt.
Checkmk ist sogesehen Multi Mandant fähig.

Jede Instanz kann auch verschiedene Versionsnummern haben.

Z.B Eine Test umgebung mit der schon Version 2 getestet wird und ne Prod die noch auf 1.6 läuft

Instanz erstellen

```
omd create <namederinstanz>
```

Beispiel

```
omd create monitoring
```

Hier bekommen wir den Hinweis, das wir zu wenig VCPUs haben.

Ist halt ne Testumgebung. Gleichzeitig sehen wir hier das Kennwort für den Webadmin

Ausgabe:

```
Creating temporary filesystem /omd/sites/monitoring/tmp...OK
```

```
Updating core configuration...
```

```
Generating configuration for core (type cmc)...
```

```
WARNING: The number of configured checkers is higher than the number of available CPUs. To avoid unnecessary context switches, the number of checkers should be limited to the number of CPUs. Recommended number of checkers: 2
```

```
Starting full compilation for all hosts Creating global helper config...OK
```

```
Creating cmc protobuf configuration...OK
```

```
Executing post-create script "01_create-sample-config.py"...OK
```

```
Restarting Apache...OK
```

```
Created new site monitoring with version 2.1.0p16.cfe.
```

```
The site can be started with omd start monitoring.
```

```
The default web UI is available at http://checkmk/monitoring/
```

```
The admin user for the web applications is cmkadmin with password: **** wird natürlich in klartext angezeigt  
****
```

```
For command line administration of the site, log in with 'omd su monitoring'.
```

```
After logging in, you can change the password for cmkadmin with 'cmk-passwd cmkadmin'.
```

Damit wäre die Installation abgeschlossen.

Wie im oberen text zu sehen.

ist das Kennwort ändern übers Terminal jederzeit möglich.

```
omd su <instanzname>
```

dann

```
cmk-passwd <nuntername>
```

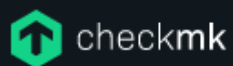
Beispiel :

```
omd su monitoring
```

```
cmk-passwd cmkadmin
```

Nun einloggen über IP/Domain Instanzname

Beispiel <https://checkmk.io/monitoring>



Username:

Password:

Login

Version: 2.1.0p16 - © [tribe29 GmbH](#)

Version #4

Erstellt: 7 Februar 2023 11:33:49 von Admin

Zuletzt aktualisiert: 9 März 2023 10:14:46 von Admin