Canonical

We are the company behind Ubuntu.

Delivered by Canonical
Mission and Philosophy - In a nutshell

Freedom to download Ubuntu - study, use, share, (re-)distribute, contribute, improve and innovate it!

Mapped to Ubuntu Server for IBM Z and LinuxONE (s390x) - the goal is:

- to expand Ubuntu’s ease of use to the s390x architecture (IBM Z and LinuxONE)
- unlock new workloads, especially in the Open Source, Cloud and container space
- to tap into new client segments
- quickly exploit new features and components - in two ways:
  - promptly supporting new hardware
  - releases built and based on the latest kernels, tool-chain and optimized libraries
- provide parity across architectures, in terms of release and feature parity and closing gaps
- provide a uniform user experience and look-and-feel
- be part of the collective world-wide Open Source power in action
- deal with upstream work and code only - no forks
- offer a radically new subscription pricing with drawer-based pricing, or alternatively provide entry-level pricing based on up to 4 IFLs
Ubuntu for IBM Z and LinuxONE - Release cadence

- **Ubuntu 16.04 L TS**
- **Ubuntu 18.04 L TS**
- **Ubuntu 20.04 L TS**
- **Ubuntu 22.04 L TS**

- **5 years base**
- **18 months**


Legend:
- Blue: in development
- Green: in service
- Yellow: ESM
- Gray: end-of-life
- Red: with s390x support
- Purple: upgrade path

- Ubuntu 16.04 LTS 5 years end-of-life
- Ubuntu 17.04 LTS with s390x support upgrade path
- Ubuntu 18.04 LTS 5 years base
- Ubuntu 19.04 LTS 5 years base
- Ubuntu 20.04 LTS 5 years base
- Ubuntu 21.04 LTS 5 years base
- Ubuntu 22.04 LTS 5 years base

- https://wiki.ubuntu.com/Releases
- https://wiki.ubuntu.com/LTS
Ubuntu Release Naming Scheme

The official Ubuntu release number is ‘xx.yy’, whereas ‘xx’ represents the year (minus 2000) and ‘yy’ the month of the release within in that year.

So Ubuntu's first release, made available in 2004 October (October: 10th month) was Ubuntu 4.10.

Since the actual release date is not known until it's ready and humans tend to prefer names rather than numbers, a set of code-names are used by developers and testers during the development phase:

<table>
<thead>
<tr>
<th>Adjective</th>
<th>Animal</th>
<th>Version</th>
<th>Adjective</th>
<th>Animal</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warty</td>
<td>Warthog</td>
<td>4.10</td>
<td>Natty</td>
<td>Narwhal</td>
<td>11.04</td>
</tr>
<tr>
<td>Hoary</td>
<td>Hedgehog</td>
<td>5.04</td>
<td>Oneiric</td>
<td>Ocelot</td>
<td>11.10</td>
</tr>
<tr>
<td>Breezy</td>
<td>Badger</td>
<td>5.10</td>
<td>Precise</td>
<td>Pangolin</td>
<td>12.04 LTS</td>
</tr>
<tr>
<td>Dapper</td>
<td>Drake</td>
<td>6.06 LTS</td>
<td>Quantal</td>
<td>Quetzal</td>
<td>12.10</td>
</tr>
<tr>
<td>Edgy</td>
<td>Eft</td>
<td>6.10</td>
<td>Raring</td>
<td>Ringtail</td>
<td>13.04</td>
</tr>
<tr>
<td>Feisty</td>
<td>Fawn</td>
<td>7.04</td>
<td>Saucy</td>
<td>Salamander</td>
<td>13.10</td>
</tr>
<tr>
<td>Gutsy</td>
<td>Gibbon</td>
<td>7.10</td>
<td>Trusty</td>
<td>Tahr</td>
<td>14.04 LTS</td>
</tr>
<tr>
<td>Hardy</td>
<td>Heron</td>
<td>8.04 LTS</td>
<td>Utopic</td>
<td>Unicorn</td>
<td>14.10</td>
</tr>
<tr>
<td>Intrepid</td>
<td>Ibex</td>
<td>8.10</td>
<td>Vivid</td>
<td>Vervet</td>
<td>15.04</td>
</tr>
<tr>
<td>Jaunty</td>
<td>Jackalope</td>
<td>9.04</td>
<td>Wily</td>
<td>Werewolf</td>
<td>15.10</td>
</tr>
<tr>
<td>Karmic</td>
<td>Koala</td>
<td>9.10</td>
<td>Xenial</td>
<td>Xerus</td>
<td>16.04 LTS</td>
</tr>
<tr>
<td>Lucid</td>
<td>Lynx</td>
<td>10.04 LTS</td>
<td>Yakkety</td>
<td>Yak</td>
<td>16.10</td>
</tr>
<tr>
<td>Maverick</td>
<td>Meerkat</td>
<td>10.10</td>
<td>Zesty</td>
<td>Zapus</td>
<td>17.04</td>
</tr>
</tbody>
</table>

The development codename of a release has a name like "Adjective Animal". Warty Warthog (Ubuntu 4.10) was the first Ubuntu release. In general, people refer to the release using the adjective, like "warty" or "breezy". The names live on in one hidden location—the archive release name in /etc/apt/sources.list and seen on the download mirror network.

Ubuntu 18.04 LTS (Bionic Beaver)

- The codename for the current LTS (Long Term Support) release 18.04 is 'Bionic Beaver' or in short 'Bionic': https://launchpad.net/ubuntu/bionic
  Release date: April, 26th 2018
- Updated major components:
  - Kernel 4.15 (linux-generic) + HWE kernels
  - Qemu-KVM 2.11.x / Libvirt (libvirt-bin) 4.0.0
  - LXD 3.0.0 (incl. clustering support)
  - GCC 7.3 → 7.4 (gcc 5, 6, 8 universe) / GDB 8.1
  - Python 3.6.5 → 3.6.7 (and 2.7.15, but not installed by default)
  - Perl 5.26
  - Ocaml 4.05
  - netplan 1.10 / netplan.io 0.36 → 0.97 (replacing ifupdown)
  - CDO ‘Queens’ (Canonical Distribution of Openstack)
  - Openssl 1.1.0.g → 1.1.1
- In order to download Ubuntu Server 18.04 LTS for IBM Z and LinuxONE, please visit: https://www.ubuntu.com/download/server/s390x
Ubuntu 18.04 LTS (Bionic Beaver)

Non-complete list of s390x-specific new features and enhancements

- improvements for IBM z14, z14 ZR1, LinuxONE Rockhopper II and LinuxONE Emperor II (1725260) (1736100)
- s390-tools major version upgrade to v2.3.0 (1735447)
- cryptsetup rebase and enhancements in support of dm-crypt (1724592)
- protected key support for dm-crypt (1741904)
- TLB enhancements (1732426) (1732452)
- TOD-Clock Epoch Extension Support (1732437) (1732691)
- DASD multi-queue (1732446) support and block layer discard support (1732440)
- Improved memory handling (1734120)
- support for new crypto hardware CEX6S (1735437)
- AP bus kernel API for KVM (1732449)
- CPU-MF/perf improvement (1735433)
- CPACF enhancements for z14 and ECC support (1724603) (1735437)
- HiperSocket connections enhancements (1735695)
- parted update for fdasd/vtoc (1737144)
- openssl-ibmca rebase (1747626)
- opencryptoki rebase for EP11 and ECC enhancement (1751272)
- lock optimization enhancement (1747877)
- libica upgrade for z14 and ECC support (1737159) and to use PRNO-TRNG to seed SHA512-DRBG (1754617)
- auto detect layer2 setting in qeth driver (1747639)
- Kernel support for STHYI/LPAR (1736093)
- rebase libpfm4 for z13/z13s CPU-MF hardware counters (1741905)

**Ubuntu 20.04 LTS (Focal Fossa)**

- The codename for 20.04 is 'Focal Fossa' or just 'Focal': [https://launchpad.net/ubuntu/focal](https://launchpad.net/ubuntu/focal)
- Ubuntu Server Long-Term Support (LTS) release
- Major components (planned):
  - Kernel 5.4
  - qemu-kvm 4.2+
  - libvirt 6.0+
  - glibc 2.31
  - binutils 2.34
  - docker 19.03.8
  - gcc 9.3 (default; gcc10 in universe)
  - gdb 9.1
  - LLVM 7,8,9,10
  - python 3.8.2 / (2.7.17 in universe)
  - golang 1.13
  - s390-tools 2.12+
  - smc-tools 1.2.2
  - openssl 1.1.1f
  - openssl-ibmca 2.1.0
  - opencryptoki 3.13.0
  - libica 3.6.1
  - qclib 2.1.0
  - apt 2.0.1
  - snapd 2.44
  - cloud-init 20.1.10
  - php 7.4+
Non-complete list of 20.04 s390x-specific new features and enhancements (since 19.10):

- Starting with Ubuntu Server 20.04 the architectural level set was changed to z13 (LP:1836907). This has a significant impact: Ubuntu Server for s90x now benefits from improved and more instructions that got introduced with z13 hardware; at the same time support for zEC12/zBC12 got dropped and the minimum supported hardware is now IBM z13 and LinuxONE Rockhopper (I) and LinuxONE Emperor (I).

- Secure Execution, a Trusted Execution Environment (TEE) for IBM Z and LinuxONE is now supported. It required adaptations in the kernel (LP:1835531), qemu (LP:1835546) and s390-tools (LP:1834534). It can only be used with IBM z15 and LinuxONE III. With Secure Execution (or the upstream name 'protected virtualization' aka 'protvirt') workloads can run in full isolation with protection for both internal and external threats, using hardware assisted key based encryption for the guest memory.

- The toolchain was significantly upgraded to gcc 9.3 - making sure that fixes like (LP:1862342) are included, even moved to gdb 9.1 (LP:1825344), that includes latest s390x hardware support - similar with LLVM, that was upgraded to v10 (LP:1853145), again to have the latest s390x hardware enhancements included (LP:1853269).

- Compression improvements got added to the kernel (LP:1830208) that allow exploitation of the hardware assisted deflate compression operation, provided by the new integrated (on-chip) compression co-processor (z15 and LinuxONE III only), by zlib and filesystems.

- The KVM virtualization stack got updated to qemu 4.2 and libvirt 6.0, and with that CPU model comparison and baselining got enabled (LP:1853315), CCW IPL support added to qemu (LP:1853316) and libvirt (LP:1853317) and several issue fixed, like (LP:1861125), (LP:1867109) and (LP:1866207). In addition KVM crypto pass-through is now included (LP:1852737), (LP:1852738) and (LP:1852744).

- Support for new CEX7S crypto express hardware (LP:1853304) and (LP:1856831) was added, as well as CPACF MSA 6 in-kernel crypto support for SHA3 (LP:1853105) and a lot of CPACF crypto co-processor (largely assembly based) optimizations and fixes in OpenSSL (LP:1853150) and (LP:1853312), incl. but not limited to ECDSA.

- Further zkey/pkey cryptography improvements were added, like extend pkey module to support AES cipher keys (LP:1830609) and (LP:1853325), enhanced handling of secure keys and protected keys: (LP:1853302) and (LP:1853303), enhancements in the zkey tools consistency checks: (LP:1853308) and (LP:1853309), a self-test of the paes cipher of paes_s390 module (LP:1854948) and fixes added, like a fix for the XTS attribute display of the validate command (LP:1862187).

- Additional cryptography and security relevant libraries got upgraded, like libica3 (LP:1853143) and openCryptoki, now with new hardware support, incl. z15 (LP:1853300), (LP:1858792), (LP:1853310), PRNO pseudo-random number support in ICA, CCA and EP11 tokens (LP:1852088) and common changes like FIPS compliant PIN encryption (LP:1854938) and crypto base movement to OpenSSL (LP:1854939). Additional fixes are incl. like fixing a failure to import ECC public keys (LP:1852089).

- Even more libraries got updated with improvements for s390x, like qclib 2.0 (LP:1852718), glibc with math library optimization (LP:1853270), Boost (LP:1694926), (LP:1859941) and (LP:1864433), but also tools, like a smc-tools update (LP:1852721), not to talk about s390-tools (LP:1834534). And additional support was added for HiperSockets Multi-Write (LP:1853292), thin provisioning DASD support (discard support for ESE volumes) (LP:1862749) and proper kprobes on ftrace (LP:1865858) on a kernel level.

- Finally zPCI enhancement, like 'zpcictl --reset' (LP:1863768) and fixes, like write through (LP:1866162) got picked up and with that not only a kernel config option change of CONFIG_PCI_NR_FUNCTIONS to 512, but also further kernel config option changes, like CONFIG_NR_CPUS and CONFIG_NUMA_EMU (LP:1864198), CONFIG_NET_SWITCHDEV (LP:1865452) and disabling HIBERNATION and PM (LP:1867753).

Please also see the official release notes: https://wiki.ubuntu.com/FocalFossa/ReleaseNotes
This is a distilled view of the 20.04.x Ubuntu Kernel Support Schedule. Depending on the installed LTS ‘point’ release, it’s either possible to use the generic and default Kernel (always until EOL) or optionally the HWE Kernel (HWE upgrade path need to be followed, starting with ‘.2’).

Starting with 20.04.2 there will be a choice of:
- 2 installation kernels (for ‘subiquity’)
- 2 target kernels (to install to disk)

After a HWE kernel opt-in, updates to next HWE are the default.

https://wiki.ubuntu.com/Kernel/LTSEnablementStack
Upgrade Path to 20.04

Ubuntu Server 19.10
Ubuntu Server 18.04 LTS

Always from latest non-LTS to current LTS and from previous LTS to current LTS.

‘do-release-upgrade’ is the recommended tool to use.

Join the webinar: “Migrating your infrastructure to Ubuntu 20.04 LTS - how, when and why?”
Blog: How to upgrade from Ubuntu 18.04 LTS to 20.04 LTS today
Wiki: https://help.ubuntu.com/community/FocalUpgrades#Ubuntu_Servers
Ubuntu Server - live installer (subiquity)

Ubuntu 20.04 LTS (Focal Fossa)

Select an image

Ubuntu is distributed on four types of images described below.

Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not install a graphical user interface.

- 64-bit PC (AMD64) server install image
  Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

- 64-bit ARM (AArch64) server install image
  For 64-bit ARMv8 processors and above.

- PowerPC64 Little-Endian server install image
  For POWER8 Little-Endian computers, such as Power Systems S894/4LC Linuxonly servers.

- IBM System z server install image
  For IBM System z series mainframes, such as IBM LinuxONE.

https://ubuntu.com/download/server/s390x

Installer update available

Version 20.03.1 of the installer is now available (19.12.1.5 is currently running).

You can read the release notes for each version at:
https://github.com/CanonicalLtd/subiquity/releases

If you choose to update, the update will be downloaded and the installation will continue from here.

[ Update to the new installer ]
[ Continue without updating ]
[ Back ]

https://ubuntu.com/server/docs/install/general
https://ubuntu.com/server/docs/install/autoinstall
Ubuntu Server - live installer (subiquity)

Use UP, DOWN and ENTER keys to select your language.

- Asturianu
- Bahasa Indonesia
- Català
- Deutsch
- English
- English (UK)
- Español
- Français
- Galés
- Hrvatski
- Latviski
- Lietuviškai
- Magyar
- Nederlands
- Norsk bokmål
- Polski
- Suomi
- Svenska
Ubuntu Server - live installer ‘Zdev’ (subiquity)
Ubuntu Server - **live installer** (subiquity)

Installer shell session activated.

This shell session is running inside the installer environment. You will be returned to the installer when this shell is exited, for example by typing Control-D or 'exit'.

Be aware that this is an ephemeral environment. Changes to this environment will not survive a reboot. If the install has started, the installed system will be mounted at /target.

```
root@ubuntu-server:~/# uname -a
Linux ubuntu-server 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10 00:21:32 UTC 2020
```

```
s390x s390x s390x GNU/Linux
root@ubuntu-server:~/# lszdev --online
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>ON</th>
<th>PERS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>zfcp-host</td>
<td>0.0.f00b</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>zfcp-host</td>
<td>0.0.f10b</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>zfcp-lun</td>
<td>0.0.f00b:0x5005076306b16b6:0x4026400600000000</td>
<td>yes</td>
<td>no</td>
<td>sdb sgl</td>
</tr>
<tr>
<td>zfcp-lun</td>
<td>0.0.f00b:0x5005076306b16b6:0x4026400600000000</td>
<td>yes</td>
<td>no</td>
<td>sda sg0</td>
</tr>
<tr>
<td>zfcp-lun</td>
<td>0.0.f10b:0x5005076306b16b6:0x4026400600000000</td>
<td>yes</td>
<td>no</td>
<td>sdd sg3</td>
</tr>
<tr>
<td>zfcp-lun</td>
<td>0.0.f10b:0x5005076306b16b6:0x4026400600000000</td>
<td>yes</td>
<td>no</td>
<td>sdc sg2</td>
</tr>
<tr>
<td>geth</td>
<td>0.0.0600:0.0.0601:0.0.0602</td>
<td>yes</td>
<td>no</td>
<td>enc600</td>
</tr>
<tr>
<td>generic-ccw</td>
<td>0.0.0009</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

```
root@ubuntu-server:~/#
```
Ubuntu Server - live installer (subiquity)

Installer shell session activated.

This shell session is running inside the installer environment. You will be returned to the installer when this shell is exited, for example by typing Control-D or 'exit'.

Be aware that this is an ephemeral environment. Changes to this environment will not survive a reboot. If the install has started, the installed system will be mounted at /target.

root@ubuntu-server:/$ lsb_release -d
Description: Ubuntu 20.04.1 LTS
root@ubuntu-server:/$ uname -a
Linux ubuntu-server 5.4.0-42-generic #46-Ubuntu SMP Fri Jul 10 00:21:32 UTC 2020
-s390x s390x s390x GNU/Linux
root@ubuntu-server:/$ snap list
Name       Version  Rev Tracking       Publisher    Notes
core18     20200724 1884 latest/stable  canonical*  base
snapt      2.45.2   8539 latest/stable  canonical*  snapd
subiquity  20.07.1+git2.5de9df3e 1969 latest/stable/... canonical* classic
root@ubuntu-server:/$
Ubuntu Server - autoinstall (user-data ‘yaml’)

```
$ cat /var/log/installer/autoinstall-user-data

# cloud-config
autoinstall:
  version: 1
refresh-installer:
  update: yes
reporting:
  builtin:
    type: print
apt:
  preserve_sources_list: false
  primary:
    - arches: [amd64, i386]
      uri: http://archive.ubuntu.com/ubuntu
    - arches: [default]
      uri: http://ports.ubuntu.com/ubuntu-ports
keyboard:
  layout: en
  variant: us
locale: en_US
identity:
  hostname: zvmguest
  password: "$6$ebJ1f8wxED22bTL4F46P0"
  username: ubuntu

user-data:
  timezone: America/Boston
users:
  - name: ubuntu
    password: "$6$KwuxED22bTL4F46P0"
    lock_passwd: false
  early-commands:
  - chzdev dasd -e 1234
network:
  version: 2
  ethernets:
    enc600:
      addresses: [10.11.12.23/24]
      gateway4: 10.11.12.1
      nameservers:
        addresses: [10.11.12.1]
  ssh:
    install-server: true
    allow-pw: true
    authorized-keys: ['ssh-rsa meQwttZ user@workstation # ssh-import-id lp:user']
⇒ /var/log/installer/autoinstall-user-data
```
Ubuntu Server Live Installer (subiquity / autoinstall)

The installation process changed with 20.04 for s390x (further improved with every point release), and the documentation was reworked and updated and can now be found here:

The landing page is the official **Ubuntu Server Guide for 20.04 LTS** (chapter 'Installation'):

Ubuntu Server Guide - 20.04 LTS:
- http: [https://ubuntu.com/server/docs/install/general](https://ubuntu.com/server/docs/install/general)
- pdf: [https://assets.ubuntu.com/v1/10d22089-ubuntu-server-guide.pdf](https://assets.ubuntu.com/v1/10d22089-ubuntu-server-guide.pdf)

The step-by-step examples from the Ubuntu Server guide about the **live installer** (subiquity) can also be found as separate documents at 'discourse', where it's possible to comment:
- **Interactive live server installation on IBM Z LPAR (s390x)**
- **Interactive live server installation on IBM z/VM (s390x)**

There also also step-by-step guides for **autoinstall**, the new way of doing non-interactive installations (succeeding preseed):
- **Non-interactive IBM Z LPAR (s390x) installation using autoinstall**
- **Non-interactive IBM z/VM (s390x) installation using autoinstall**
HW compression (NXU) support in Ubuntu 20.04

- Ubuntu Server 20.04 LTS advantages:
  - Hardware assisted compression supported is built-in.
  - Hence tools like gzip/gunzip, tar -czf, compression in IBM Java 8 SR6+, and everything that uses zlib (since it’s a user space instruction) - even your kernel decompress after each boot - gets a nice speed up out of the box.
  - Significant speed-ups of 20x to 40x (zlib/DEFLATE) - for free (on z15)!
- Latest supported hardware compression function is DEFLATE, which is supported by default with Ubuntu 20.04 (s390x), too: `CFLAGS="-O2 -DDFLTCC and -DDFLTCC_LEVEL_MASK=0x7e"` is used (means hardware acceleration compression is enabled for compression levels 1-6).
- If unsure check with:
  
  ```
  $ strings /usr/bin/gzip | grep DFLTCC
  DFLTCC
  $ strings /usr/lib/s390x-linux-gnu/libz.so* | grep DFLTCC
  DFLTCC
  ```
Ubuntu 22.04 (Jammy Jellyfish)

- The codename for 22.04 is 'Jammy Jellyfish' or simply 'Jammy': [https://launchpad.net/ubuntu/jammy](https://launchpad.net/ubuntu/jammy)
- Ubuntu Server LTS aka long term support release
  - Final Release: Apr, 21st 2022 (Release Candidate: Apr 14th 2022, Beta Mar 31st 2022)
- Major components:
  - Kernel 5.15
  - qemu 6.2
  - libvirt 8.0.0
  - virt-manager 4.0.0
  - glibc 2.35
  - binutils 2.38
  - gcc-default 11.2 (12, 10, 9 in universe)
  - gdb 12.0
  - LLVM 14 default (11, 12, 13 and 15 exp.)
  - python 3.10
  - go / golang 1.18
  - ruby 3.0
  - valgrind 3.18.1
  - smc-tools 1.7.0
  - openssl 3.0.2
  - openssl-ibmca 2.2.3
  - libica 4.0.1
  - opencryptoki 3.17+
  - cryptsetup 2.4.3
  - cloud-init 22.1
  - docker.io 20.10.12
  - netplan 1.10.1
  - util-linux 2.37
  - PHP 8.1
  - qclib 2.3.0
  - systemctl 249
- New packages/libraries:
  - libzpc 1.0.0
  - libzdnn 0.4.0
  - s390-tools 2.20+
Non-complete list of 22.04 s390x-specific new features and enhancements (since 21.10):

- Like with every new Ubuntu release (respectively its kernel) the s390-tools package needs to be upgraded, this time to v2.17 (Bug: 1929024), including zdsfs transparent data set conversion, allowing Linux to transparently read and write EBCDIC-encoded data sets as ASCII (Bug: 1926749), the integration option for the zkey repository into an enterprise key-management system with a KMIP interface (Bug: 1932521) and changes in the Secure Execution Header defaults for plaintext control flags, PCF (Bug: 1932177).
- In addition to moving to gcc 11.2 as default, further tool-chain updates were made, like updating to LLVM 13 (that derived s390x optimizations from 12 (Bug: 1926709)) and z15 support (Bug: 19329184).
- And more predictable interface names for RoCE adapter were introduced, which requires kernel (Bug: 1929185) as well as systemd changes (Bug: 1929184).
- Several KVM enhancements specific to s390x were picked up, like performance improvements due to Spinlock Yield Forwarding (Bug: 1905021) and allowing KVM to let SIE interpret specification exceptions (Bug: 1932157), adding support to indicate secure (execution) guests (Bug: 1933173) and improved persistence in vfio-ccw device assignments in libvirt (Bug: 1887929).
- Another area of improvements is cryptography. With the upgrade of opencryptoki to 3.16 (Bug: 1928767) cca token import and export of secure key objects is now supported (Bug: 1913301), ep11 token support for attribute bound keys (Bug: 1913303) and ep11 token protected key support (Bug: 1914215) got added.
- And with the upgrade to libica v3.8.0 (Bug: 1928799) there are now software fallback calls to openSSL/libcrypto (Bug: 1929176). cryptsetup got upgraded too, to v2.3.6 (Bug: 1929046), as well as openssl-ibmca to v2.2.0 (Bug: 1929052), that now makes the ibmca engine call libica without software fall backs (only register ibmca functions if libica confirms it as hardware function) and let ibmca do the fallback (Bug: 1929175).
- On the kernel level AP bus and zcrypt uevent extensions were added to the zcrypt driver (Bug: 1933496) and CEX8 toleration included (Bug: 1933805).
- In addition preparation were included in the kernel (Bug: 1932174) and qemu (Bug: 1932175) for new IBM Z hardware.
- The Query Capacity library (qclib) got bumped to its latest version 2.3.0 (Bug: 1926586), the upgraded glibc v2.34 library comes with several s390x related improvements (Bug: 1927079), similar with the binutils update to v2.37 (Bug: 1927080). On top zlib received BRCM2 optimization for s390x (Bug: 1932177) and various library updates, like qcclib, glibc, PCRE2 and Eigen, Leading to further performance improvements on s390x; as well as the brand new (s390x-only) libraries: libzpc and libzDNN.

Please also see the official release notes: [https://discourse.ubuntu.com/t/jammy-jellyfish-release-notes](https://discourse.ubuntu.com/t/jammy-jellyfish-release-notes)
What’s a point(-release)?
Regular respin and hardware enablement for 2+ years

- **What** is a Point Release?
  Ubuntu LTS point releases provide users with a new kernel (except “.1”) as well as a roll up of previous package updates and security patches. In total 5 point releases are made available per LTS release.

- **Goals** (as outlined in the [Ubuntu Point Release Process](https://wiki.ubuntu.com/PointReleaseProcess))
  - Refresh hardware support in LTS releases for carefully-selected hardware
  - Roll up accumulated stable updates into updated images to reduce download requirements for new deployments
  - Maintain stability of existing installations

- This nowadays 10 year old blog post on 'The Art of Release' (by Mark Shuttleworth) is still relevant today, covers a brief summary of point-releases, and finally shows Canonicals reliable release history over the last decade:
  "We also committed, for the first time, to a regular set of point releases for 8.04 LTS. These will start three months after the LTS, and be repeated every six months until the next LTS is out. These point releases will include support for new hardware as well as rolling up all the updates published in that series to date. So a fresh install of a point release will work on newer hardware and will also not require a big download of additional updates."

https://wiki.ubuntu.com/Releases
https://wiki.ubuntu.com/PointReleaseProcess
http://www.markshuttleworth.com/archives/146
Ubuntu LTS ‘point’ Releases
Regular respin and hardware enablement for 2+ years

The ‘point’ releases include support for new hardware (starting with .2 with an optional HWE Kernel, that’s available in addition to the default and GA kernel), as well as rolling up all the updates published in that series to date. So a fresh install of a point release will work on newer hardware and will also not require a big download of additional updates.

https://wiki.ubuntu.com/Releases
http://www.markshuttleworth.com/archives/146
IBM z16 and LinuxONE Emperor 4 & Canonical Ubuntu

IBM: We built the powerful and secure platform for business.
Canonical: We built our best Open Source Ubuntu LTS distribution.

Make the most out of your zNext by using Ubuntu Server 22.04 LTS for IBM zSystems.

- IBM z16 hardware support added to Ubuntu kernel, drivers, core libraries and tool, math. libraries and tool-chain.
- Renewal of the entire crypto stack, incl. CryptoExpress 8S, openssl 3, libica4, openssl-ibmca, openCryptoki, cryptsetup
- Updated tool-chain, incl. optimizations and support with gcc 11.2, gdb 12 / binutils 2.38, LLVM 14 and Go 1.18
- New network capabilities: HSCI Multi-MAC and SMC Enterprise ID, statistics and SMC-R v2 support.
- QEMU/KVM virtualization stack and Secure Execution updates, optimized existing and added some new libraries.
Ubuntu Server Certified Hardware (s390x)

- **z16 A01**
  - LPAR (DPM & classic)
  - z/VM

- **z15 T01 / z15 T02**
  - LPAR (DPM & classic)
  - z/VM

- **z14 M01-M05 / z14 ZR1**
  - LPAR (DPM & classic)
  - z/VM

- **z13 / z13s**
  - LPAR (DPM & classic)
  - z/VM

- **zBC12 / zEC12**
  - LPAR
  - z/VM

- **LinuxONE Emperor 4 LA1**
  - LPAR (DPM & classic)
  - z/VM

- **LinuxONE III / LinuxONE LT2**
  - LPAR (DPM & classic)
  - z/VM

- **LinuxONE Emperor II / Rockhopper II**
  - LPAR (DPM & classic)
  - z/VM

- **LinuxONE Emperor / Rockhopper**
  - LPAR (DPM & classic)
  - z/VM

LPAR certifications cover KVM too, since KVM is integral to Ubuntu Server.

https://certification.ubuntu.com/certification/server/models/?query=&vendors=IBM&release=16.04+LTS
https://certification.ubuntu.com/certification/server/models/?query=&vendors=IBM&release=18.04+LTS
https://certification.ubuntu.com/certification/server/models/?query=&vendors=IBM&release=20.04+LTS
IBM Z and LinuxONE - Tested platforms

IBM tested and Partner certified Linux environments
IBM has tested and certified Linux environments of distribution partners. You can review the statements of the individual Linux distribution for each hardware. Check the statements of the individual release for each hardware. Some are out of service, and extended support may be available. Please contact your distribution partner. You can obtain a Support Line contract for remote technical support or a contract with a third-party provider.

Certified by Linux partner
For detailed version levels see the information on the original site. Overview shows Linux distributions in service. Extended support is available for Linux distributions that are out of service.

<table>
<thead>
<tr>
<th></th>
<th>z16™</th>
<th>z15™</th>
<th>z14® (all models)</th>
<th>z13®</th>
<th>z13s®</th>
<th>zEnterprise®</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emperor 4</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LinuxONE III</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emperor II</td>
<td>●</td>
<td>●</td>
<td>Rockhopper II</td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rockhopper</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zEC12, zBC12</td>
<td>●</td>
<td>●</td>
<td></td>
<td>●</td>
<td></td>
<td>●</td>
</tr>
</tbody>
</table>

Ubuntu 22.04

Ubuntu 20.04

Ubuntu 18.04

Ubuntu 16.04

Please see more details and all footnotes at:
https://www.ibm.com/it-infrastructure/z/os/linux-tested-platforms
http://www.ibm.com/systems/z/os/linux/resources/testedplatforms.html
**Where to run Ubuntu Server on Z and LinuxONE**

- **Logical Partition (LPAR)**
- **z/VM guest**
- **KVM virtual machine**
- **Container (on an Ubuntu host)**

- IBM zEC12/zBC12, z13*/z13s* and newer resp. the equivalent LinuxONE systems and zPDT
- HMC in classic or DPM mode

* requires Ubuntu Server 20.04 LTS or newer
Deploying Ubuntu Server on Ubuntu KVM

A Linux installation on KVM is similar (if not equal) for all Linux platforms, incl IBM Z and LinuxONE. However the tooling can be more or less convenient - here are the options provided by Ubuntu:

- **kvm** - kvm-enabling command-line wrapper for qemu-system-<arch>
- **virsh** - command-line management user interface for KVM (and other hypervisors)
- **virt-manager** - graphical management user interface for KVM (and other hypervisors)
- **virt-inst** - cli tools to provision new KVM (and other) virtual machines, part of virt-tools
- **uvt-kvm** - part of the uv-tools, Ubuntu virtualisation front-end for libvirt and KVM
- **LXD** - the scope of LXD v4.2 was expanded to KVM, now beyond container management
- **CDO** - Charmed Distribution of OpenStack with Nova KVM support
- **multipass** - get an instant Ubuntu VM with a single command ([https://multipass.run/](https://multipass.run/))

Depending on the tool and needs the virtual machines may be installed by:

- booting with the installer kernel and initrd
- directly booting from the ISO image or
- booting the installer over the network via PXE boot

Installations can be interactive using ‘d-i’ or non-interactive using ‘preseed’.

In addition **Cloud images** are available and can directly be started (no need to install) by:

- downloading the Cloud image manually and starting it with for example virsh
- or using uvt-simplestreams-libvirt to just get and sync it from the image archive
One-Liner example for 18.04 / bionic - already released:

$ NAME=bionic; uvt-simplestreams-libvirt sync release=$NAME label=release arch=$(arch) && uvt-kvm create $NAME-vm release=$NAME label=release && uvt-kvm wait $NAME-vm && uvt-kvm ssh $NAME-vm

Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-54-generic s390x)

System information as of Sun Jul 14 13:44:22 UTC 2019

... To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.

ubuntu@bionic-vm:~$ Notice that ‘daily’ images are not produced for all Ubuntu releases, just for those that are in active development.

One-Liner example for 19.10 / eoan - currently in development / daily:

$ NAME=eoan; uvt-simplestreams-libvirt sync release=$NAME label=daily arch=$(arch) && uvt-kvm create $NAME-vm release=$NAME label=daily && uvt-kvm wait $NAME-vm && uvt-kvm ssh $NAME-vm

LXD (>=4.2+) and Multipass (>=1.6.2 with LXD)

$ snap info lxd
$ sudo snap install lxd
$ sudo lxd init   # --auto

YouTube video: LXD on an IBM mainframe by Stephane Graber

https://multipass.run

$ snap info multipass
$ sudo snap install multipass

https://linuxcontainers.org
# Canonical Platform Summary

<table>
<thead>
<tr>
<th>Cloud App Ecosystem</th>
<th>IoT App Ecosystem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Framework</strong> JAAS</td>
<td><strong>EdgeX</strong> EdgeX</td>
</tr>
<tr>
<td><strong>Juggle</strong></td>
<td><strong>Snap Store</strong> Snap Collection</td>
</tr>
<tr>
<td><strong>JUJU ROCKS</strong></td>
<td><strong>SnapD</strong> SnapD snappy</td>
</tr>
<tr>
<td><strong>VM Appliances</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Charmhub</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Identity</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Charmed OSM</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Kubeflow</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Base IT</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Observability</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Platform</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroK8s &amp; Charmed</td>
</tr>
<tr>
<td><strong>MicroK8s</strong></td>
</tr>
<tr>
<td>SnapD snappy</td>
</tr>
<tr>
<td>Core</td>
</tr>
<tr>
<td>KVM</td>
</tr>
<tr>
<td>Bootloader</td>
</tr>
</tbody>
</table>

| **Hardware** |ubuntu |
Cloud Solution Stack: Enabling Operations at Scale

**Landscape:**
Systems management & patching

**LXD:**
Pure-container hypervisor

**Ubuntu OpenStack:**
Canonical-produced optimized and interop tested openstack packages
Enterprise-class, hyperscale server operating system

**Ubuntu Server (incl. KVM):**
Enterprise-class, hyperscale server operating system

**Containers (LXD):**
Cloud deployment, integration, scaling, upgrading

**VMs (KVM):**
KVM VMs

**Juju:**
Cloud deployment, integration, scaling, upgrading

**OpenStack:**
Canonical-produced optimized and interop tested openstack packages
Enterprise-class, hyperscale server operating system

**Containers (Docker):**
Kubernetes: Container coordination

**Docker:**
Kubernetes: Container coordination

**MAAS (KVM):**
‘Metal-as-a-Service’ on s390x for KVM VM provisioning only

**LPAR**
Ubuntu - The #1 Platform for Containers

Canonical’s Distribution of Kubernetes
Pure Kubernetes tested across the widest range of clouds with modern metrics and monitoring, brought to you by Canonical.

Docker Engine on Ubuntu
Docker Engine is a lightweight container runtime with robust tooling that builds and runs your containers. Over 65% of all Docker-based scale out operations run on Ubuntu.

LXD
A pure-container hypervisor that runs unmodified Linux guest operating systems with VM-style operations.

We help enterprises run containers at scale, on public, private and bare metal clouds.
Juju - The Services Modeling Tool
Brings all our Open Source Packages and IBM Software

Several OSS Charms & Bundles have been enabled for POWER and Z, where the code base got ported.

- MySQL
- MariaDB
- OpenStack
- RabbitMQ
- Wordpress
- HaProxy
- MemCache
- Kubernetes …

https://juucharms.com/q/?tags=ibm
https://jaas.ai/u/ibmcharmers
https://jaas.ai/openstack-base/bundle

Open source. Solution-driven.
Model, build and scale your environments on any cloud.

Browse the store →
IBM zSystems and LinuxONE Hypervisors

- LPAR
  - PR/SM | DPM
- IBM
  - z/VM
- KVM
  - lxc | KVM
- LXD
  - lxc | KVM
IBM Z & LinuxONE - Juju Deployment Options

- ‘manual cloud ’ = list of pre-installed hosts
  - LPARs, z/VM guests, KVM VMs, LXD containers, ...

- ‘local Cloud ’ = LXD
  - before: Cloud experience inside a pre-installed Ubuntu host automatically using LXD containers
  - now with LXD v3: Cloud experience cross pre-installed Ubuntu hosts on a low latency network

- MAAS ‘bare metal’ ¹ or KVM (former MAAS Pods) ²
  - Cloud experience cross pre-defined Ubuntu hosts
  - integration with other platforms managed by MAAS

¹ MAAS for LPARs requires HMC in DPM mode
² MAAS KVM support for s390x available since 2.5.3, recommended is using 2.6.x
Juju - Allows Reuse Across Clouds

- Local System (LXD or KVM)
- Test & Dev VMs
- commercial Hypervisors (VMware)
- Bare Metal (MAAS)
- commercial Public Clouds (AWS, GKE, Azure, etc.)
- OpenStack

https://docs.jujucharms.com/clouds
Canonical Distribution of OpenStack (CDO)

Management & Automation
- Landscape & Autopilot
- Juju

Infrastructure Services
- Nova
- Horizon
- Ceilometer / Telemetry
- Keystone
- Neutron
- Swift
- Cinder
- Glance

Ubuntu Server 20.04 LTS / 22.04 LTS

MAAS
- Intel
- IBM Power
- IBM LinuxONE
- IBM Z
Canonical's *Ubuntu Cloud archive* allows users the ability to install newer releases of Ubuntu OpenStack on an Ubuntu Server as they become available up through the next Ubuntu LTS release.

The Ubuntu OpenStack support lifecycle is as follows:
CK - Charmed Kubernetes

Canonical's Charmed Distribution Of Kubernetes (CDK):
https://jujucharms.com/canonical-kubernetes/
https://jaas.ai/charmed-kubernetes

Canonical expands enterprise container portfolio:

Components of Charmed Kubernetes 1.20
https://ubuntu.com/kubernetes/docs/1.20/components
CK Infrastructure Example using LXD

LXD example, here with 10 systems total

business / user workload
runtime for user workload
kubernetes infrastructure, here CDK
machine / system infrastructure, here LXD but can be: LPAR, KVM, MAAS, public Clouds, OpenStack, etc.
Kubernetes/CK - juju status (cli)

juju status or better
watch -c juju status --color
indicates that the deployment is fine.
Nothing marked in red (or yellow),
no workload states error or blocked.
Kubernetes/CK Juju GUI - applications
MicroK8s

Low-operation, minimal production Kubernetes, for development, Cloud, clusters, workstations, servers, Edge and IoT.

- Smallest, fastest, fully-conformant Kubernetes that tracks upstream releases and makes clustering (optional) trivial.
- Default single node or optionally multi-node cluster setup possible.
- Goal is to eliminate everyday administration from Kubernetes clusters. Install, cluster, and then just watch it fly - one may alter MicroK8s configuration, but many people don’t bother.
- Automatic data store, API services and leader election.
- MicroK8s also runs in an immutable container, so your Kubernetes itself is fully containerised.
- MicroK8s can update automatically, with rollback on failure.
- Defaults to the most widely used Kubernetes options, hence it ‘just works’ with no additional config needed.
- Try it out (Howto) - and provide some feedback (https://github.com/ubuntu/microk8s)!

$ sudo snap install --classic
Why Canonical Kubernetes?

- Pure upstream, latest & greatest versions
- 100% compatible with Google’s Kubernetes
- Operates on AWS, Azure, GCE, OpenStack, VMWare, LXD, KVM, ...
- Secured. TLS, (Kernel Live patching), confinement
- Upgradable between each Kubernetes Release
- Cost effective at scale
- Bare metal operations with MAAS (tbd, today KVM only on s390x)
1. Manage your environment
2. Discover & manage your network
3. Manage your resources
4. Configure your hardware
5. Install your operating system
### MAAS DPM/LPAR and KVM - Example Environments

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Region- and Rack-Controller)</td>
<td>LPAR TA04</td>
<td>LPAR TA05</td>
<td>LPAR TA06</td>
</tr>
<tr>
<td>IBM Z and LinuxONE (HMC in DPM mode)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Managing LPARs only.

### Managing LPARs and KVM VMs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Region- and Rack-Controller)</td>
<td>LPAR TA04</td>
<td>LPAR TA05</td>
<td>LPAR TA06</td>
</tr>
<tr>
<td>IBM Z and LinuxONE (HMC in DPM mode)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

KVM Host

Managing LPARs and KVM VMs.
MAAS - IBM Z and LinuxONE

- Ability to build heterogeneous cloud including IBM Z managed by MAAS
- Full private cloud experience with MAAS + Juju
- Cross LPARs / servers experience (Cloud regions and HA zones)
- Resources allocated only when needed
- Benefits from IBM Z scale-up functionality

MAAS LPAR (DPM)
1. Manage LPARs from MAAS on Z systems running in DPM mode
2. Pre-define LPARS, but register LPARs as ‘machines’ to MAAS
3. OS deployment into LPARs using MAAS UI or API (e.g. using Juju)

MAAS KVM (Pods)
1. Prepare or MAAS deploy an LPAR that can act as KVM host
2. Configure KVM (virsh or LXD) and register KVM in MAAS
3. KVM VMs are created and VMs deployed using MAAS UI or API

More efficient use of resources by dynamically allocating hardware!
The **API** of MAAS is what provides the most value.
MAAS on IBM Z or LinuxONE - Resources

- How do I configure and use IBM Z with MAAS?  
  [Link](https://maas.io/docs/snap/3.0/ui/power-management#heading--configure-use-ibm-z)

- MAAS 3.0 - What’s New - IBM Z DPM/LPAR  
  [Link](https://maas.io/docs/snap/3.0/ui/whats-new-in-maas#heading--ibm-z-dpm)

- MAAS on Discourse  
  [Link](https://discourse.maas.io)

- MAAS on IBM Z - Show and tell (by Lee Trager)  
  [Link](https://drive.google.com/file/d/1MZyhLL1znKeq4ARWePcywDOg1Wzeu4SY/view)
  [Link](https://people.canonical.com/~fheimes/maasz/MAAS%20on%20IBM%20Z.mkv)
3 Complementary Automation Tools

**MAAS***
- PHYSICAL and KVM PROVISIONING
- DYNAMIC RE-PURPOSING

**JUJU**
- SERVICES MODELING, DEPLOYMENT
- SCALING

**LANDSCAPE**
- ADMINISTRATION
- AUDIT
- COMPLIANCE

* s390x: MAAS KVM only
Abstract
This article summarizes our experiences with the setup, configuration and usage of OpenSSL, PKCS#11 and its related components for exploiting hardware-assisted cryptographic operations on IBM LinuxONE and IBM Z for clear key operations. The required steps are described, as well as findings in the areas of performance improvement using OpenSSH, Apache HTTP server and IBM Java. Based on our positive experiences we recommend that you should make use of these capabilities whenever performing cryptographic workloads on Ubuntu Server for IBM Z and IBM LinuxONE.

The paper is available via IBM Techdocs WP102721 ‘Hardware cryptographic support for IBM Z and IBM LinuxONE with Ubuntu Server’: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102721
Hardware cryptography with Ubuntu on s390x

Enable hardware assisted cryptography support on Ubuntu Server for s390x with a few easy steps (same for all Ubuntu Server for s390x releases)

For Ubuntu Server >= 22.04 that uses openssl 3 (it’s different for 21.10 and earlier!) do:

Install the required packages for the hardware crypto support:
```
sudo apt-get install libica-utils libica? openssl-ibmca
```

Optional: Create a backup of the default openssl configuration file:
```
sudo cp -p /etc/ssl/openssl.cnf{,_$(date +%Y-%m-%d_%H:%M:%S).backup}
```

Replace the existing OpenSSL configuration file with the one that is included as sample (one line):
```
sudo cp /usr/share/doc/openssl-ibmca/examples/openssl.cnf.sample /etc/ssl/openssl.cnf
```

Alternatively, one may re-generate a sample and copy this one over:
```
ibmca-engine-opensslconfig  # generates: ./openssl.cnf.ibmca
...
sudo cp ./openssl.cnf.ibmca /etc/ssl/openssl.cnf
Hardware cryptography with Ubuntu on s390x

What did we get: "openssl engine -c"

```
$ openssl engine
(dynamic) Dynamic engine loading support
(ibmca) Ibmca hardware engine support

$ openssl engine -c
(dynamic) Dynamic engine loading support
(ibmca) Ibmca hardware engine support

[RSA, DSA, DH, RAND, DES-ECB, DES-CBC, DES-OFB, DES-CFB, DES-EDE3, DES-EDE3-CBC,
DES-EDE3-OFB, DES-EDE3-CFB, AES-128-ECB, AES-192-ECB, AES-256-ECB, AES-128-CBC,
AES-192-CBC, AES-256-CBC, AES-128-OFB, AES-192-OFB, AES-256-OFB, AES-128-CFB,
AES-192-CFB, AES-256-CFB, id-aes128-GCM, id-aes192-GCM, id-aes256-GCM, SHA1, SHA256,
SHA512, ED25519, ED448, X25519, X448]

$ openssl ciphers -s -v -stdname
TLS_AES_256_GCM_SHA384 TLS_AES_256_GCM_SHA384 TLSv1.3 Kx=any Au=any
Enc=AESGCM(256) Mac=AEAD
... # about 30 cipher suites will be listed here
```
Hardware cryptography with Ubuntu on s390x
What does “icainfo” show on z15 CPACF + CEX7S + Ubuntu Server 20.04 LTS

## Cryptographic algorithm support

<table>
<thead>
<tr>
<th>function</th>
<th>hardware</th>
<th>dynamic</th>
<th>static</th>
<th>software</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-1</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-224</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-256</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-384</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-512</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-512/224</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA-512/256</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>SHA3-224</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>SHA3-256</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>SHA3-384</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>SHA3-512</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>SHAKE-128</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>SHAKE-256</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>GHASH</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>P_RNG</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>DRBG-SHA-512</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>ECDH</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>ECDSA Sign</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>ECDSA Verify</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>ECKGEN</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Ed25519 Keygen</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Ed25519 Sign</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Ed25519 Verify</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Ed448 Keygen</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>RSA ME</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>RSA CRT</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>DES ECB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>DES CBC</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>DES OFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>DES CFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>DES CTR</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES ECB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES CBC</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES OFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES CFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES CTR</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>3DES CMAC</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES ECB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES CBC</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES OFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES CFB</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES CTR</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES CMAC</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES XTS</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>AES GCM</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

No built-in FIPS support.
Pervasive encryption: Protecting data at rest
Optimistic Usage of zkey in Ubuntu Server (subiquity live) Installer (20.04 or newer)

- Protecting data at rest in the context of Pervasive Encryption is very popular and well documented:
  - Pervasive Encryption for Data Volumes: [HTML](#) or [PDF](#)
- It is straightforward, but requires manual steps to setup - even just for supplemental (data) volumes.

- But it becomes much more challenging in case the **root** filesystem (and **swap**) should be encrypted the same way!

- But here is where the **optimistic usage of zkey** of the **installer** (subiquity) of **Ubuntu Server 20.04.1** (or higher) simplifies this **root and swap encrypted setup** tremendously!

- **pre-reqs** for the optimistic usage of zkey in Ubuntu’s installer (debian-installer aka d-i) is:
  - CryptoExpress adapter (5S or higher) with at least one domain
  - an initial master key configured (either with TKE or the 'IBM CCA Host Libraries and Tools')
  - and either ECKD/DASD or an zFCP/SCSI disk storage
Pervasive encryption: Protecting data at rest
Optimistic Usage of zkey in Ubuntu Server (subiquity live) Installer (20.04 or newer)
**Pervasive encryption: Protecting data at rest**

Optimistic Usage of zkey in Ubuntu Server (subiquity live) Installer (20.04 or newer)

### Storage configuration

#### FILE SYSTEM SUMMARY

<table>
<thead>
<tr>
<th>MOUNT POINT</th>
<th>SIZE</th>
<th>TYPE</th>
<th>DEVICE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>31.490G</td>
<td>new ext4</td>
<td>new LVM logical volume</td>
</tr>
<tr>
<td>/boot</td>
<td>1.000G</td>
<td>new ext4</td>
<td>new partition of multipath device</td>
</tr>
</tbody>
</table>

#### AVAILABLE DEVICES

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>TYPE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ubuntu-vg (new, encrypted)</td>
<td>LVM volume group</td>
<td>62.980G</td>
</tr>
<tr>
<td></td>
<td>free space</td>
<td>31.490G</td>
</tr>
</tbody>
</table>

[ Create software RAID (md) ]

[ Create volume group (LVM) ]
Pervasive Encryption: Protecting Data in Flight
Different approaches and use cases

- OpenSSL and libcrypto:
  - de-facto standard TLS and crypto libraries used by many projects, no IBM Z specific configuration required
  - exploitation of IBM Z CPACF and SIMD code by libcrypto (w/o ibmca engine)
  - focus on TLS 1.2 and 1.3 ciphers
  - support for z14 AES-GCM accepted for OpenSSL version 1.1.1

- IPsec:
  - transparently uses CPACF through the in-kernel crypto API
  - Kernel 4.15 and later use new CPACF instruction for AES-GCM

- IBM Java 8 / JCE (Partner Archive)
  - IBM Java 8 service refresh 5 and later use z14 CPACF instructions
  - exploitation of IBM Z CPACF and SIMD code
Secure Boot (aka Secure IPL)

- Secure boot attributes to the Pervasive Encryption effort
- The IBM z15 and LinuxONE III hardware introduce secure boot (for SCSI IPL): requires a Kernel 5.3+ and s390-tools 2.9 (rec. 2.11) → Ubuntu 20.04 LTS
- HMC’s Load task and Activation Profile come with a new check-box: ‘Enable Secure Boot for Linux’ in case ‘SCSI Load’ is selected.
- For Linux (on s390x) two new sysfs entries got introduced:
  - `/sys/firmware/ipl/has_secure` - “1” indicates hw support for secure boot, otherwise “0”
  - `/sys/firmware/ipl/secure` - “1” indicates that secure IPL was successful, otherwise “0”
- zipl bootloader supports secure-boot with the “--secure” argument (0: secure boot disabled, 1: enabled, auto: enabled if environment supports secure boot)
- Ubuntu signs the kernel and the stage3 part of zipl bootloader (using X.509)
- Ubuntu Server on s390x defaults to secure-boot (starting with 19.10) - in case the underlying environment supports it and ‘SCSI Load’ is used.
- Secure boot is supported by the new Ubuntu live installer (subiquity).
The general idea behind secure execution is to **protect data in-use**. It's a firmware based Trusted Execution Environment (TEE), that provides support for full isolation of KVM guests using hw assisted guest memory encryption and state protection. Protection is provided against guest data corruption and theft, bad and malicious console usage, bad and malicious hypervisor administrators and even buggy or compromised hypervisors and with that it's helpful to achieve compliance, especially for Cloud service providers. Allows customers to run their critical / sensitive workloads in house or in Clouds with the same maximum level of privacy and protection - since even admins can't access the data! The general idea: If you are unsure if you can guarantee or trust the hypervisor, an **Ultravisor** is needed -- the Ultravisor is largely based on firmware and uses special hardware instructions. Hardware z15 LinuxONE III (with FC 115 - free of charge) and kernel, qemu and (s390-)tools support. → Ubuntu 20.04 LTS is the first release that supports Secure Execution!
Inherent Ubuntu Security Features

General settings - not platform specific

For example:

- fstack Protector (gcc)
- Heap Protector (glibc)
- Pointer Obfuscation (glibc)
- ASLR types (Stack, libs/mmap, exec, BRK, VDSO) (kernel)
- Built as PIE (gcc)
- Built with Fortify Source (gcc)
- Built with -fstack-clash-protection (gcc)
- 0-address protection (kernel)
- /dev/mem protection (kernel)

https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project/Recommended_Settings

https://wiki.ubuntu.com/Security/Features
ESM - Extended Security Maintenance

Extended Security Maintenance provides ongoing security fixes for Ubuntu LTS, for the Linux kernel and essential packages beyond the 5-year basic maintenance: [https://ubuntu.com/esm](https://ubuntu.com/esm)

Ubuntu ESM is available for Ubuntu Advantage for Infrastructure (UA-I) Essential, Standard and Advanced customers. (Subscriptions for IBM Z and LinuxONE are always UA-I Advanced). Ubuntu Advantage for Infrastructure covers physical servers, virtual machines, containers (and desktops).

Existing UA customers can retrieve their credentials through the Ubuntu Advantage portal: [https://ubuntu.com/advantage](https://ubuntu.com/advantage)

ESM continues security updates for:

- high and critical CVEs (Common Vulnerabilities and Exposures), in the Ubuntu base OS
- with Ubuntu LTS (14.04 and) 16.04 for up to 3 years after the end of base support
- with Ubuntu LTS 18.04 and later (*until further announcement*) for up to 5 years.

[https://ubuntu.com/advantage](https://ubuntu.com/advantage)
[https://ubuntu.com/security/esm](https://ubuntu.com/security/esm)
ESM - Consistent Lifecycle

- Ubuntu 20.04 LTS
- Ubuntu 18.04 LTS
- Ubuntu 16.04 LTS

- Orange boxes represent Hardware and maintenance updates.
- Gray boxes represent Interim release Standard Support.
- Dark purple boxes represent Extended Security Maintenance (ESM).

Timeline:
- 2014
- 2016
- 2018
- 2020
- 2022
- 2024
- 2026
- 2028
- 2030
Why Ubuntu Server LTS on s390x?

- LXD included
- release parity
- 5 years base support
- drawer-based pricing
- MAAS DPM/LPAR
- no 31-bit legacy
- various repos/archives: Partner, UCA, PPAs
- easy availability and trial
- same L&F like on other platforms
- LTS & non-LTS mix
- PE support

- OSS software stack currency
- OpenStack integration
- size and scope of the repositories
- Ubuntu KVM
- LXD
- Ubuntu MAAS
- MAAS provider
- LXD client
- Juju manual, MAAS grow
- Ubuntu Landscape client
- drawer-based pricing

- Why Ubuntu Server LTS on s390x?
- ‘point’ releases with refreshed ISO and Cloud images
- new software every 2 months (non-LTS)
- virtualization options: LPAR, z/VM, KVM, LXD + more container
- Multipass

- IBM Java
- Linux Kernel
- snap
- latest toolchain
- latest arch. level set to EC12+

- Any Questions?
Thank you!

Thanks a lot - and stop by at:
https://ubuntu-on-big-iron.blogspot.com
Why Ubuntu on IBM Z and LinuxONE - In a Nutshell

Use the most wide-spread Linux OS not only on your workstation, but also on your IBM Z Server and get the most out of your IBM Z server with current and close to upstream packages, tool-chain and kernels, one of the broadest Linux package archives and a cost efficient way to run your workloads based on a new drawer-based pricing option.

In addition see this non-exhaustive list of Ubuntu on IBM Z's unique features and values:

- Arch. level set of zEC12 - compiled for zEC12 and higher; exploiting all zEC12 features by default
- Point Releases - Ubuntu LTS releases with regular package updates and updated ISO images
- Parity - Release and feature parity across all platforms, includes same look-and-feel
- Two kernels - for stability (default) or exploitation of latest IBM Z hardware facilities and features (HWE)
- Snaps - platform and distribution agnostic packaging system on top of debs, perfectly suited for ISVs
- ZFS - feature-rich file-system with snapshot capabilities, provided as native Kernel module
- Juju/Charms - for service orchestration; modelling, deployment, scaling and upgrade, even of bundles of services
- LXD - allows to handle containers like full virtual machines, but with (machine) container performance
- CDK - Canonical's Distribution of Kubernetes; infrastructure runs on KVM or LXD, container on docker or containerd
- CDO - Canonical's Distribution of OpenStack, supporting single and cross LPAR scenarios with KVM and LXD
- Cloud and LXD images - pre-packages and ready to to use KVM Cloud and LXD container images
- Deployment - broad options with LPAR (DPM & classic), IBM z/VM, Ubuntu KVM, LXD/LXC, Docker, CDO, CDK
- Security Certifications - for FIPS 140-2, Common Criteria (EAL2), DISA STIG and CIS
- Ubuntu Advantage - Infrastructure, UA-I (Enterprise-level support for Ubuntu Server on IBM Z, incl. CDO, CDK)
Documentation - get involved

https://discourse.ubuntu.com

https://ubuntu.com/server/docs
Cloud Native & Confidential Computing on IBM Z & LinuxONE with Ubuntu 20.04 (webinar)