ubuntu

Ubuntu Server

for IBM Z and LinuxONE

What's New - June 2021

Frank Heimes, Tech. Lead Z, Canonical Ltd.

Ubuntu on Big Iron: ubuntu-on-big-iron.blogspot.com

CANONICAL

Canonical

We are the company behind Ubuntu.

ubuntu

Delivered by Canonical

Ubuntu Server for IBM Z and LinuxONE (s390x)

Mission and Philosophy - In a nutshell

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

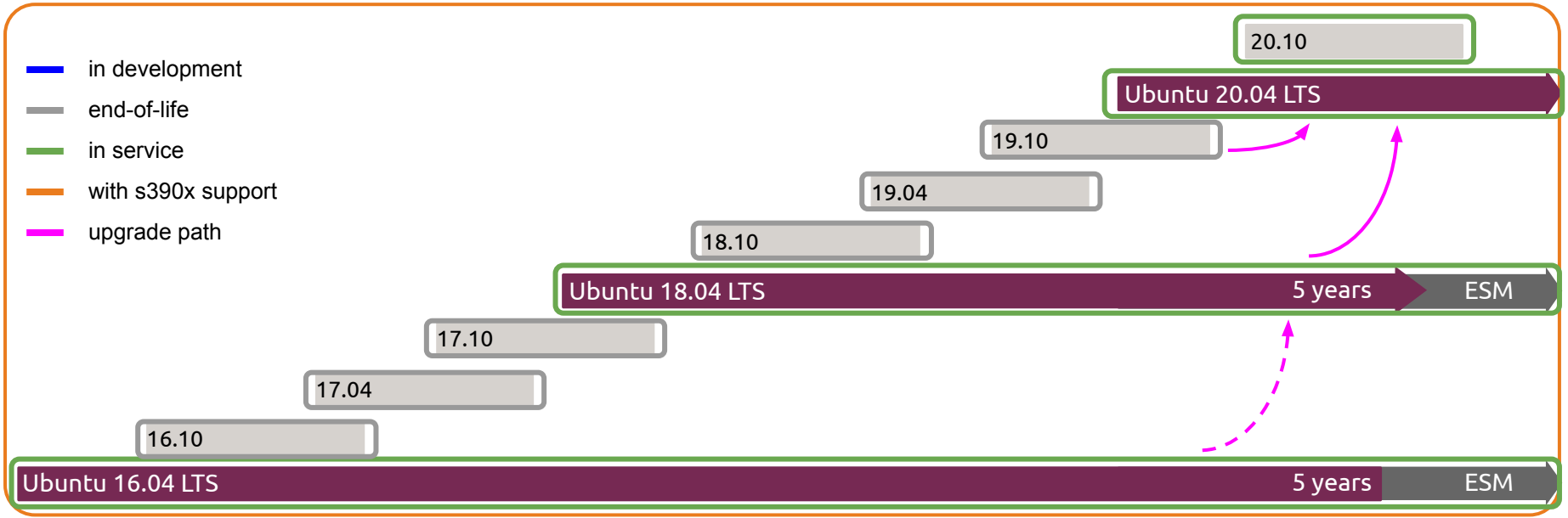
Mapped to Ubuntu Server for IBM Z and LinuxONE (s390x) - the goal is:

- 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

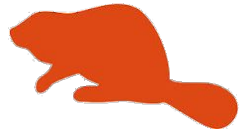
Release Cadence - Ubuntu

https://wiki.ubuntu.com/Releases
https://wiki.ubuntu.com/LTS
https://en.wikipedia.org/wiki/List_of_Ubuntu_releases

16.04 16.10 17.04 17.10 **18.04** 18.10 19.04 19.10 **20.04** 20.10 21.04



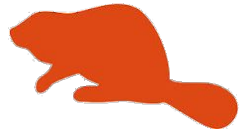
Ubuntu 18.04 LTS (Bionic Beaver)



- The codename for the current LTS (Long Term Support) release 18.04 is 'Bionic Beaver' or in short 'Bionic': https://launchpad.net/ubuntu/bionic
- Bionic Release Schedule: https://wiki.ubuntu.com/BionicBeaver/ReleaseSchedule
  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
  - docker.io 17.12.1 → 18.09.5
  - Open vSwitch 2.9 → 2.9.2
  - cloud-init 18.2.14 → 19.1.1
  - MongoDB 3.6.3
  - Postgresql 10+
  - Redis 4.0.9
  - chrony 3.2 (replacing ntpd)
  - glibc (libc-bin) 2.27
  - s390-tools 2.3.0
  - llvm 6.0
- In order to download Ubuntu Server 18.04 LTS for IBM Z and LinuxONE, please visit: https://www.ubuntu.com/download/server/s390x

Ubuntu Server

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 AES acceleration (1735438) (1735436)
- HiperSocket connections enhancements (1735691)
- parted update for fdasd/vtoc (1737...)
- 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)

- more hw specific support,
- better hw exploitation,
- initial introduction of general zkey / protected key support
- further enhancements and new crypto features (in regards to PE)

Ubuntu Server

For an overall 18.04 release description, please see the official release notes: https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes

Ubuntu 20.04 LTS (Focal Fossa)

- The codename for 20.04 is 'Focal Fossa' or just 'Focal': https://launchpad.net/ubuntu/focal
- Ubuntu Server Long-Term Support (LTS) release
- Release Schedule: https://wiki.ubuntu.com/FocalFossa/ReleaseSchedule
  Final Release: Apr, 23rd 2020 (Release Candidate: Apr 16th 2020, Beta Apr 2nd 2020)
- Release Notes: https://wiki.ubuntu.com/FocalFossa/ReleaseNotes (s390x-specifics)
- 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+

Ubuntu Server 20.04 (Focal Fossa)

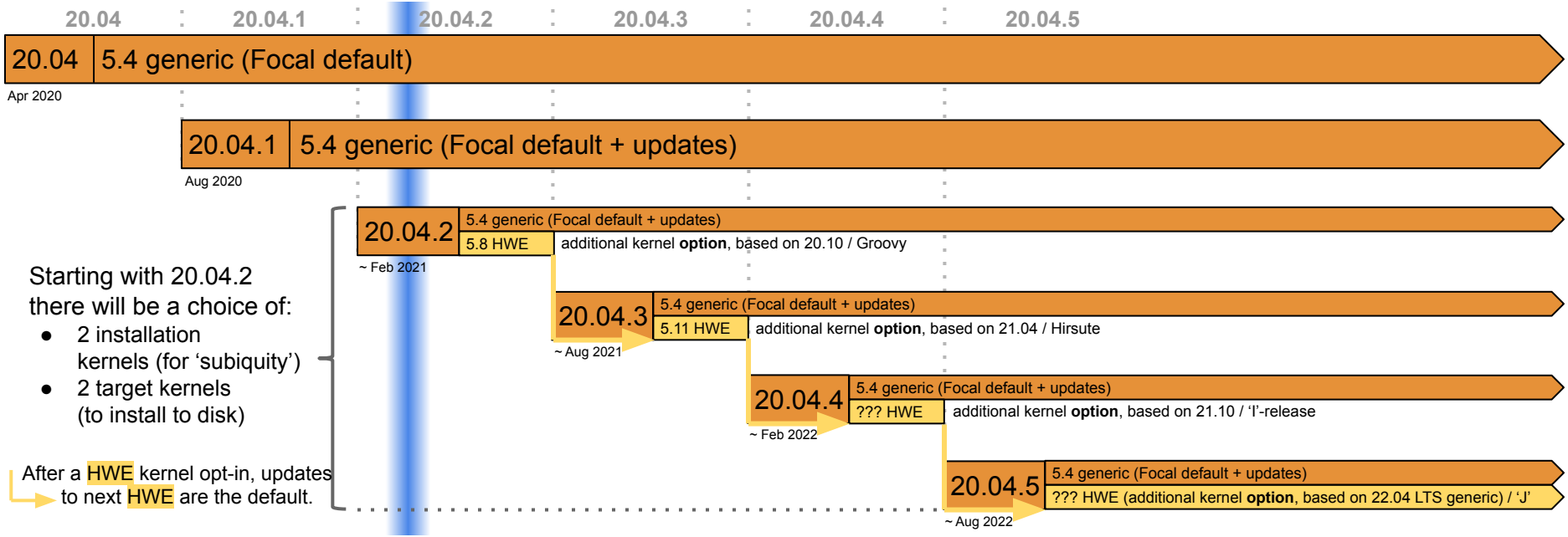
Non-complete list of 20.04 s390x-specific new features and enhancements (since 19.10):

- change of minimal architectural requirement to z13, with that zEC12/zBC12 support got dropped
- expanded hardware support for z15 in general, on-chip compression (deflate), CPACF MSA 6 and CryptoExpress CEX7 and several libraries (OpenSSL, libica3, opencryptoki, qclib, etc.)
- several zkey/pkey cryptography improvements, with regard to pervasive encryption
- Secure Execution, a Trusted Execution Environment (TEE), support for full isolation of KVM guests using hardware assisted guest memory encryption
- in addition virtualization stack updates of qemu-kvm, libvirt, incl. crypto passthrough and more
- tool-chain upates to gcc 9.3, gdb 9.1, LLVM 10, and key library updates incl. crypto libraries and glibc
- several kernel optimizations and kernel config adjustments
- subiquity is the new default installer for Ubuntu Server for s390x
- and with that 'autoinstall' supersedes 'preseed'
- Secure boot (for SCSI IPL) and IPL from NVMe (20.04.2)

Please also see the official release notes: https://wiki.ubuntu.com/FocalFossa/ReleaseNotes

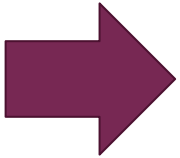
Ubuntu 20.04.x LTS Kernel Support Schedule

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’).



https://wiki.ubuntu.com/Kernel/LTSEnablementStack
https://wiki.ubuntu.com/Kernel/LTSEnablementStack#Kernel.2FSupport.A18.04.x_Ubuntu_Kernel_Support

Upgrade Path to 20.04



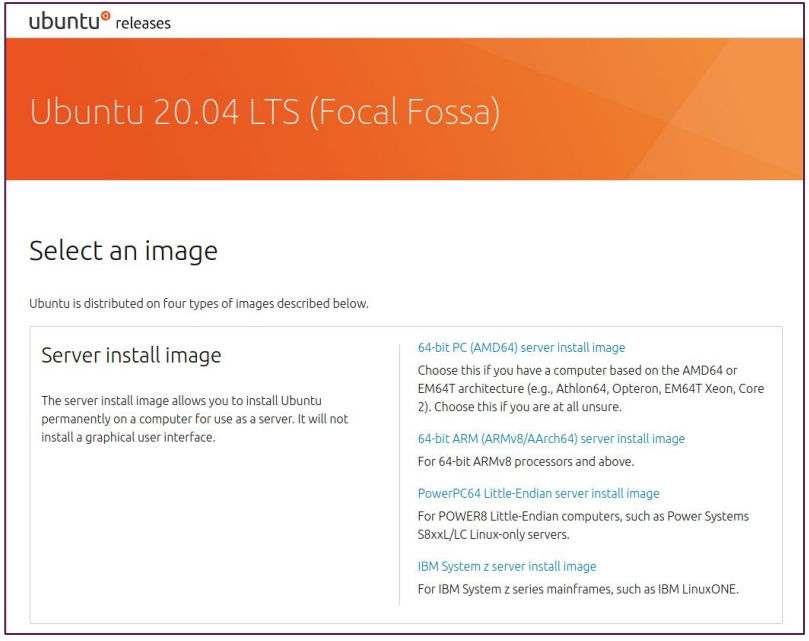
Ubuntu Server 19.10
Ubuntu Server 18.04 LTS

Ubuntu Server 20.04.x LTS

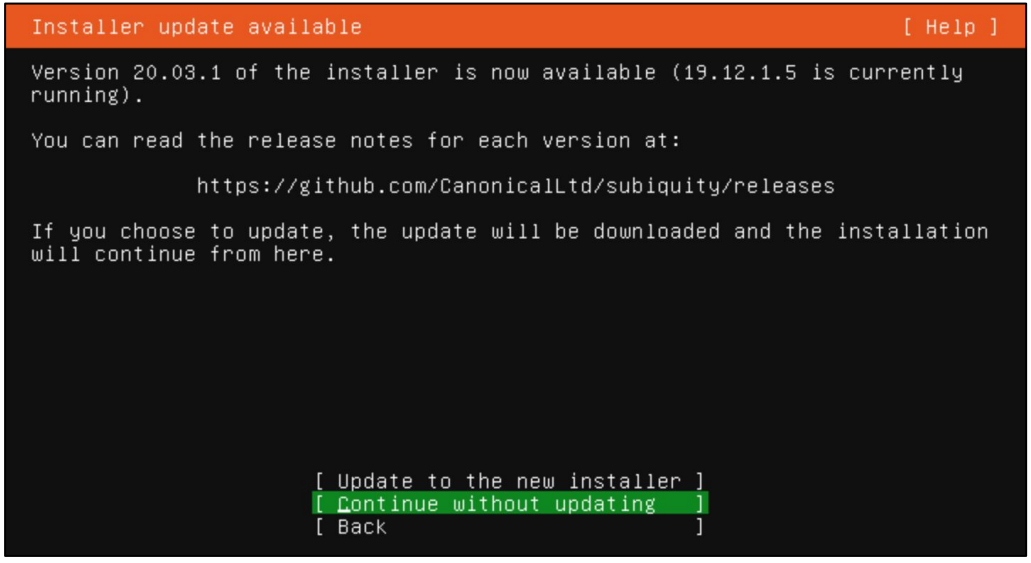
Always from latest non-LTS to current LTS and from previous LTS to current LTS.
'do-release-upgrade' is the recommended tool to use.

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)

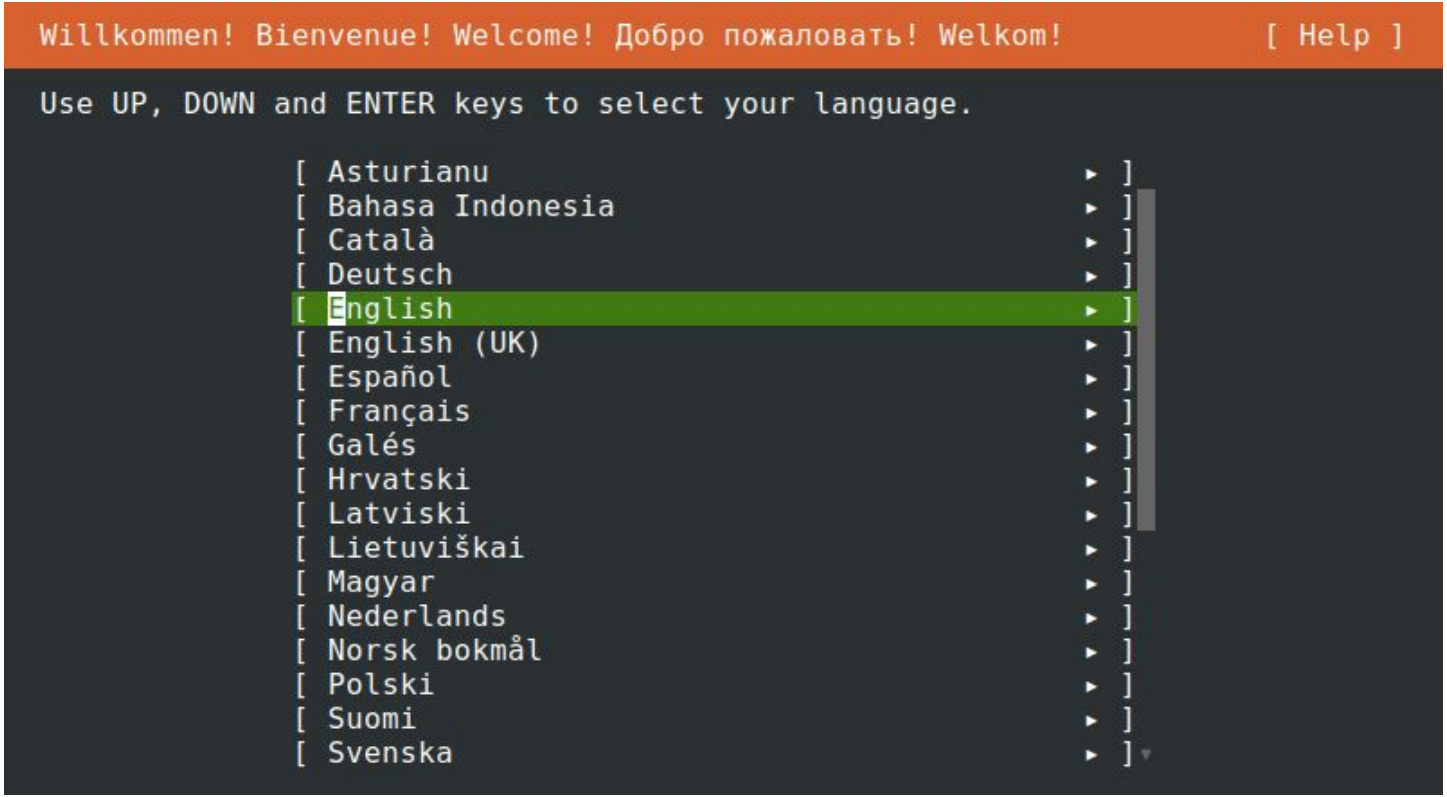


https://ubuntu.com/download/server/s390x

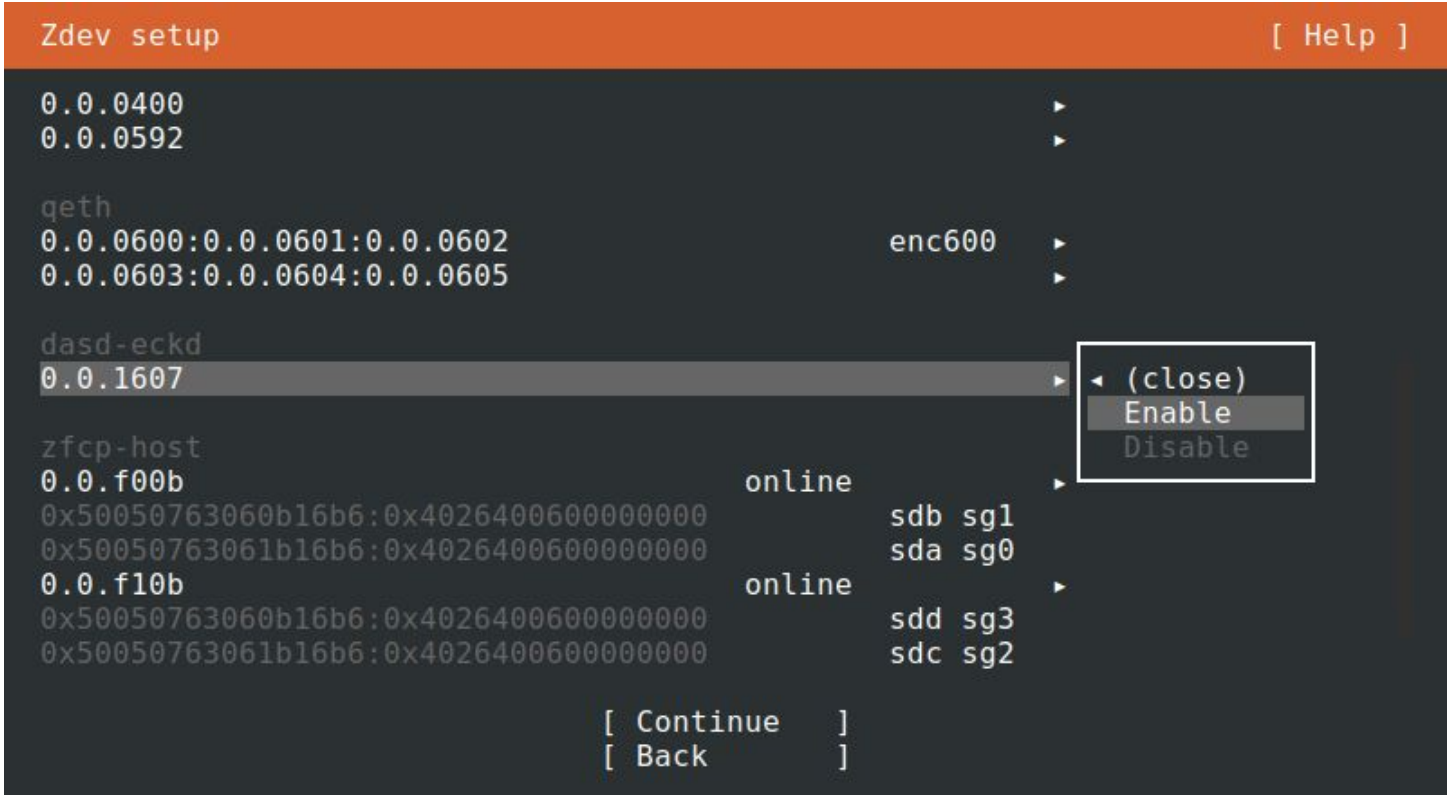


https://ubuntu.com/server/docs/install/general
https://ubuntu.com/server/docs/install/autoinstall

Ubuntu Server - live installer (subiquity)



Ubuntu Server - live installer (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
TYPE         ID                                                 ON   PERS  NAMES
zfcp-host    0.0.f00b                                           yes  yes
zfcp-host    0.0.f10b                                           yes  yes
zfcp-lun     0.0.f00b:0x50050763060b16b6:0x4026400600000000     yes  no    sdb sg1
zfcp-lun     0.0.f00b:0x50050763061b16b6:0x4026400600000000     yes  no    sda sg0
zfcp-lun     0.0.f10b:0x50050763060b16b6:0x4026400600000000     yes  no    sdd sg3
zfcp-lun     0.0.f10b:0x50050763061b16b6:0x4026400600000000     yes  no    sdc sg2
qeth         0.0.0600:0.0.0601:0.0.0602                         yes  no    enc600
generic-ccw  0.0.0009                                           yes  no
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
snapd      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 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 1f00
  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  meQwtZ
user@workstation # ssh-import-id lp:user']
```

⇒ /var/log/installer/autoinstall-user-data

Ubuntu Server Live Installer (subiquity)

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
- 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 buit-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 hw compression funtion 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 21.04 (Hirsute Hippo)

- The codename for 21.04 is 'Hirsute Hippo' or simply 'Hirsute': https://launchpad.net/ubuntu/hirsute
- Ubuntu Server non-LTS aka development release
- Release Schedule: https://discourse.ubuntu.com/t/hirsute-hippo-release-schedule/18539
  Final Release: Apr, 22nd 2021 (Release Candidate: Apr 15th 2021, Beta Mar 29th 2021)
- Release Notes: https://discourse.ubuntu.com/t/hirsute-hippo-draft-release-notes/19221
- Major components:
  - Kernel 5.11
  - qemu: 5.2+
  - libvirt: 7.0.0
  - glibc 2.33
  - binutils 2.36.1
  - gcc 10.2.1 default (7, 8, 9, 11 in universe)
  - gdb 10.1
  - LLVM 12 default (11, 13 in universe)
  - python 3.9.2 / (2.7.18 in universe)
  - go / golang 1.16
  - valgrind 3.16.1
  - wireshark 3.4.4
  - openblas 0.3.13
  - s390-tools 2.16.0
  - smc-tools 1.5.0
  - openssl 1.1.1j
  - openssl-ibmca 2.1.1
  - opencryptoki 3.15.1+
  - libica 3.7.0
  - apt 2.2.1
  - snapd 2.49+
  - cloud-init 21.1
  - docker.io 20.10.2
  - netplan 1.10.1
  - util-linux 2.36.1
  - qlibc 2.2.1
  - subiquity Installer Improvements:
    - update on the fly
    - NVMe IPL
    - DASD FBA
    - ModA (n * Mod1) etc.
  - Link time optimization (LTO) '-flto'

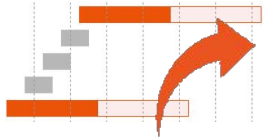
Ubuntu Server 21.04 (Hirsute Hippo)

Non-complete list of 21.04 s390x-specific new features and enhancements (since 20.10):

- Network improvements with SMC-D v2, new smc-tools, SMC-R Link Group support, wireshark support for SMC, and Hipersockets/Ethernet Converged Interfaces for z/OS Hipersocket L2 connectivity.
- Virtualization stack updates and features, like enhanced diagnose data, full zPCI function properties implementation, virtio-fs, libvirt node device driver support for DASD and vfio-ap matrix device.
- General storage enhancements, like IPL Load Normal, stand-alone dump and log DASD EDIF support.
- s390x imporvements in various packages, like Valgrind (z14 support), binutils, OpenBLAS or qclib.
- Broad cryptography updates, with zkey integration of EKMF stage1, zkey LUKS2 enhancements, zcrypt device driver improvements, EP11 related enhancements for pkey and zkey, Opencryptoki version bump with patches on top, PKCS #11 3.0 baseline provider support, enhanced EP11 token functionality and improved key management tool support for key deletion.
- Finally several installer enhancements (also in 20.04.2), like DASD FBA, DASD ECKD pass over via virtio-blk, low-level DASD ECKD format, DASD ECKD ModA EAV/EAV-II support and refinements in LVM.

Please also see the official release notes: https://discourse.ubuntu.com/t/hirsute-hippo-release-notes

What's a point(-release)?
Regular respin and hardware enablement for 2+ years

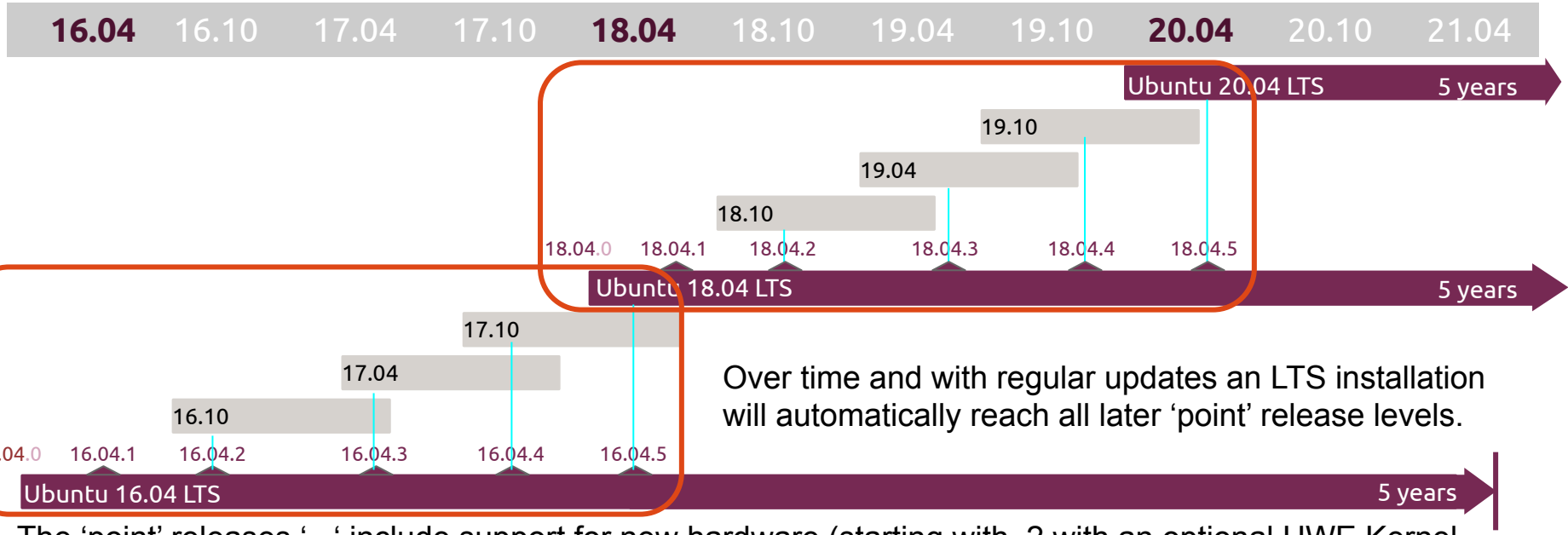


- **What** is a Point Release?
  Ubuntu LTS point releases provide users with a new kernel (except ".1") as well as a roll up of previous package updates and security patches. In total 5 point releases are made available per LTS release.
- **Goals** (as outlined in the Ubuntu Point Release Process)
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



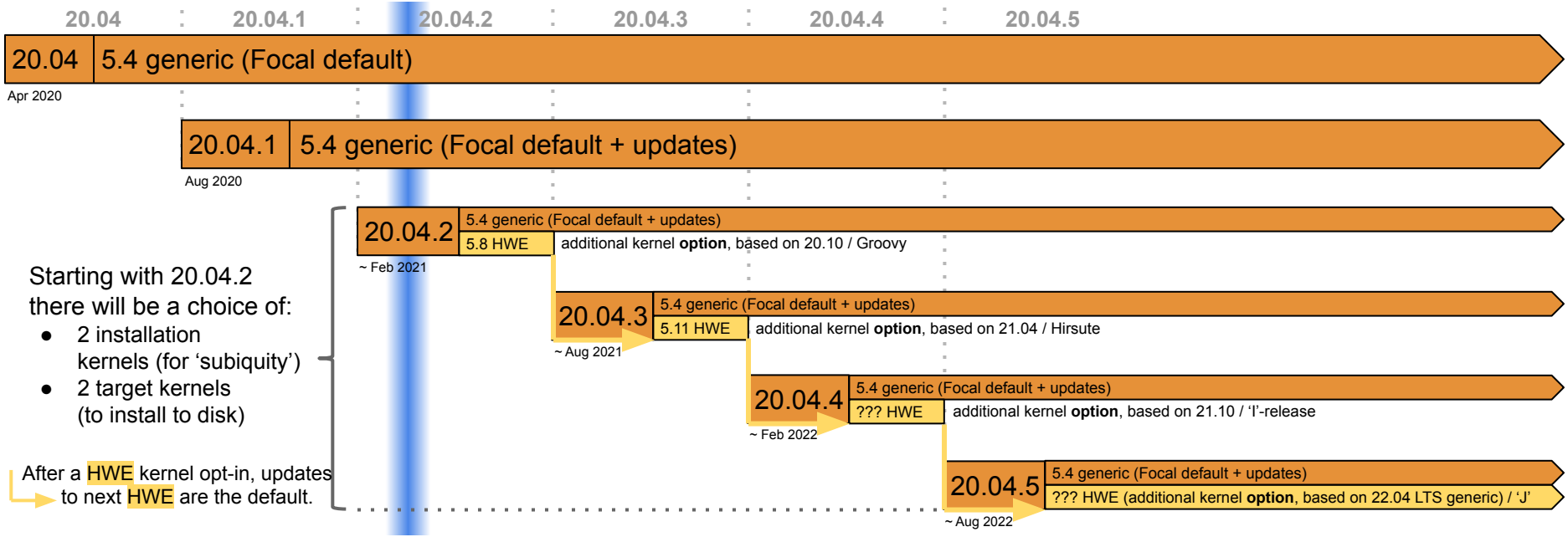
Over time and with regular updates an LTS installation will automatically reach all later ‘point’ release levels.

The ‘point’ releases ‘.x‘ 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

Ubuntu 20.04.x LTS Kernel Support Schedule

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’).



https://wiki.ubuntu.com/Kernel/LTSEnablementStack
https://wiki.ubuntu.com/Kernel/LTSEnablementStack#Kernel.2FSupport.A18.04.x_Ubuntu_Kernel_Support

Ubuntu Server Certified Hardware (s390x)

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 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 intergral 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.

	z15 LinuxONE III	z14 M0x Emperor II	z14 ZR1 Rock-hopper II	z13 Emperor	z13s Rock-hopper	zEnterprise — zEC12, zBC12
Ubuntu 20.04	●	●	●	●	●	
Ubuntu 18.04	●	●	●	●	●	●
Ubuntu 16.04	●	●	●	●	●	●

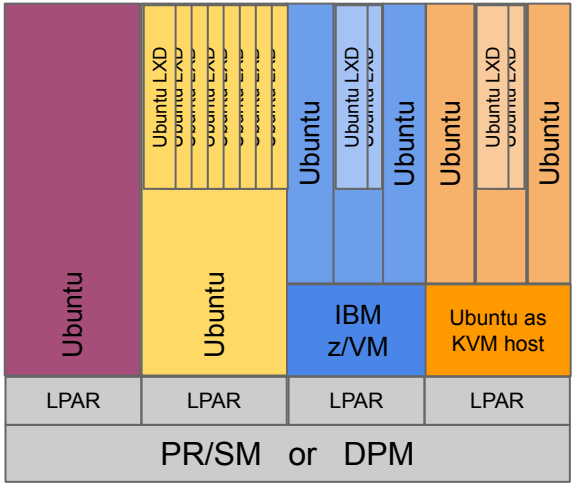
[Table got cut to incl. Ubuntu OS only and the hardware that is supported by Ubuntu.]

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 s390x?

Ubuntu Server for s390x runs:

- **'native'** in **LPAR**
  on IBM Z this is as close as possible to bare metal
- as 'IBM **z/VM**' **guest**
  a guest aka virtual machine running on IBM's z/VM hypervisor
- as **KVM virtual machine** on an **Ubuntu host**
  using plain Ubuntu Server
- as **Container** on an Ubuntu host
  using **LXD**, **lxc, Docker**, **kubernetes/k8s/CDK***
  Containers can be combined with any of the above options
- on **zEC12**, zBC12**, z13, z13s, z14 M01-M05, z14 ZR1, z15 T01/T02, LinuxONE Emperor / Rockhopper, LinuxONE Emperor II / Rockhopper II, LinuxONE III** and even on **zPDT**



- in **classic** or **DPM** (Dynamic Partition Manager) mode

* The Charmed Distribution Of Kubernetes: https://jaas.ai/canonical-kubernetes
** zEC12', zBC12* are not supported by 20.04 and higher anymore

http://insights.ubuntu.com/2016/04/22/ubuntu-16-04-lts-for-ibm-linuxone-and-ibm-z-systems-is-now-available/

ubuntu Certified

Ubuntu Server

Deploying Ubuntu Server on s390x

It is virtually the same for all Linux for IBM Z aka s390x distributions:

- Recommendation is to setup an **FTP install server** - not only for installation itself, but also for:
  - backup purposes
  - z/VM maintenance and service
  - z hardware maintenance, service and backup (iocp)
  - can be used for multiple install options, not only LPAR
  - can be used for multiple Ubuntu releases, multiple Linux distributions and different architectures
  - can be used to enable non-interactive installations (store *preseed* files there)
  - allows to do almost everything from remote (once properly setup) !
- physical CDROM/DVDROM installation - inserted into HMC (LPAR)
- USB installation - inserted into HMC (LPAR)
- boot local installer kernel and installer initrd (z/VM - IPL, KVM - virsh, virt-install)
- boot form ISO image (KVM - virsh, virt-install <using --cdrom or --location>, optionally with preseed)
- PXE netboot (KVM - virsh, optionally with preseed; DPM LPAR)
- debootstrap (used to install a Linux in a system without using an installation disk, also for chroot envs.)
- direct use without the need to install:
  - Ubuntu Cloud image (KVM - uvtools, OpenStack - customization via cloud-init)
  - Ubuntu container image (LXD, lxc, Docker, CDK/kubernetes)

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 - Canonical's Distribution of OpenStack with Nova KVM support
- **multipass** - get an instant Ubuntu VM with a single command (https://multipass.run/)

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

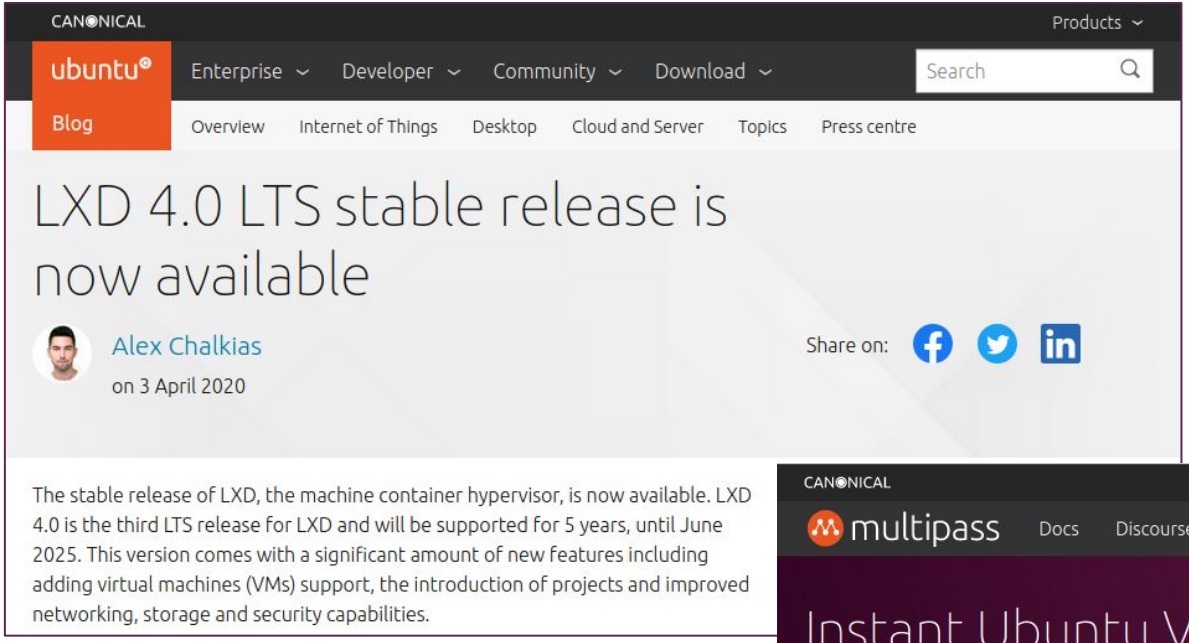
- booting with the installer kernel and initrd
- directly booting from the ISO image or
- booting the installer over the network via PXE boot

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:

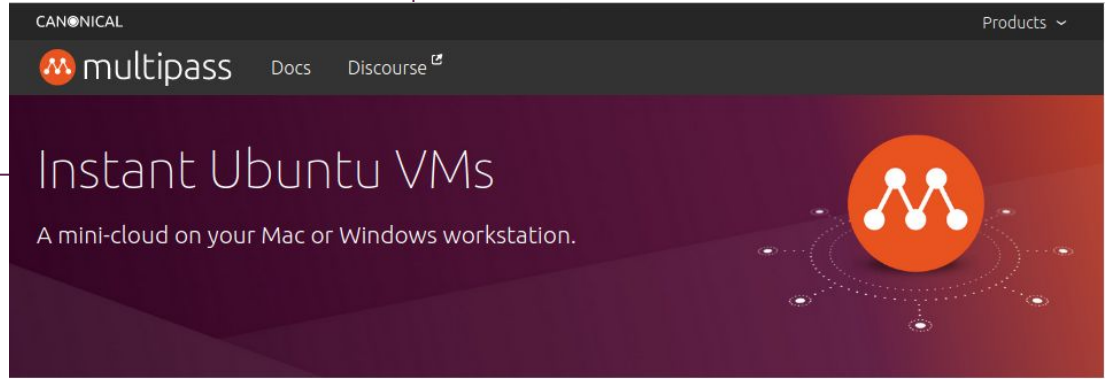
- downloading the Cloud image manually and starting it with for example virsh
- or using uvt-simplestreams-libvirt to just get and sync it from the image archive

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

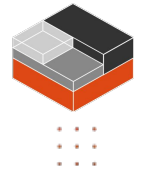


https://linuxcontainers.org

https://multipass.run

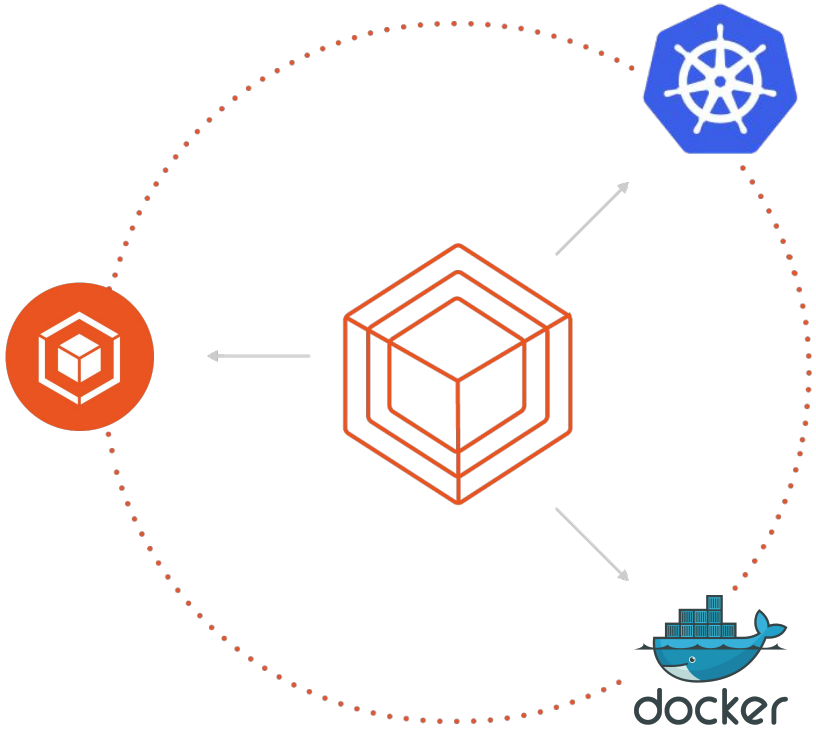


Ubuntu - The #1 Platform for Containers



LXD

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



Canonical's Distribution of Kubernetes

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

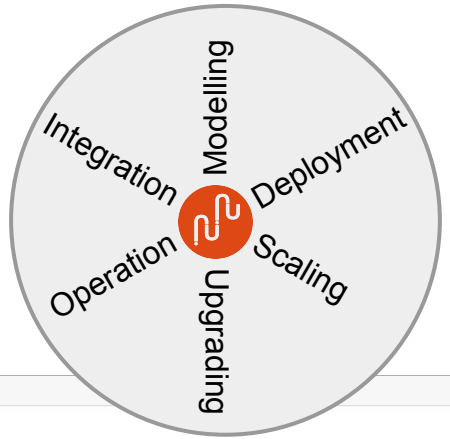
Docker Engine on Ubuntu

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.**

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

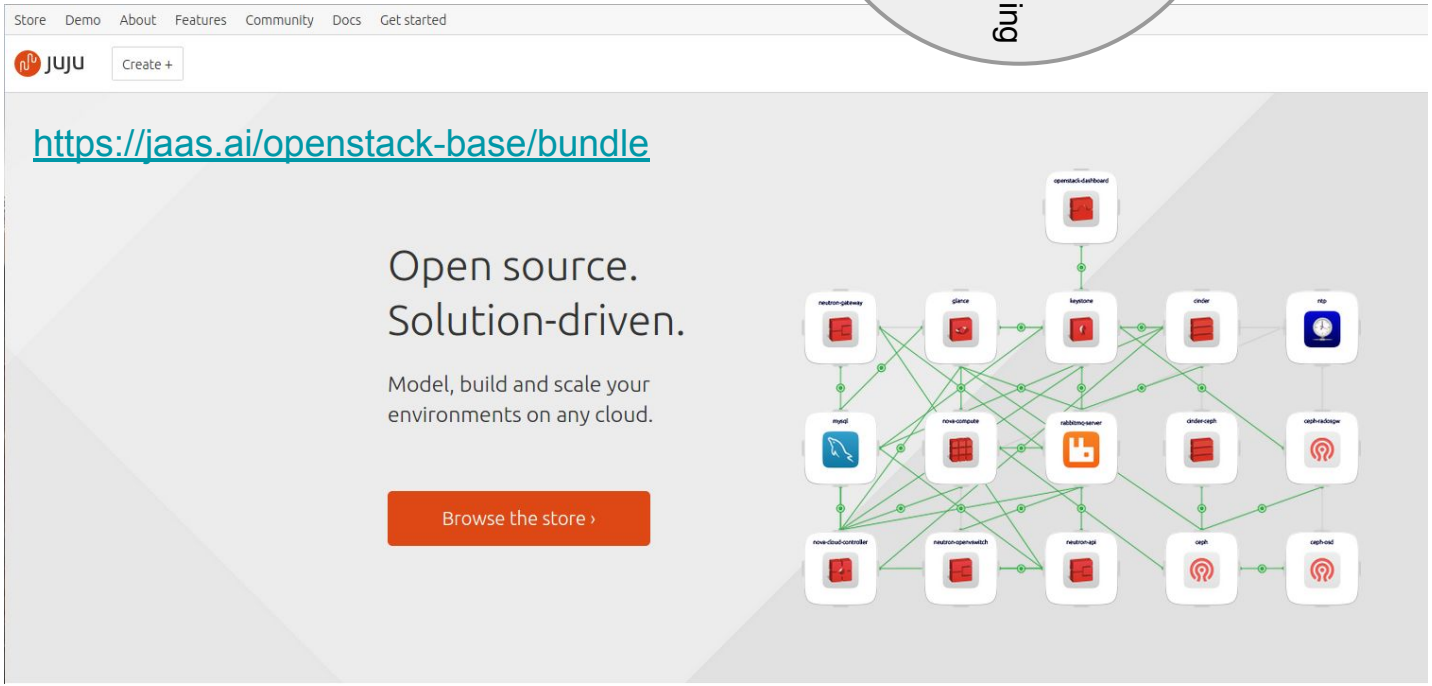


Several OSS Charms & Bundles have been enabled for POWER and Z, where the code base got ported.

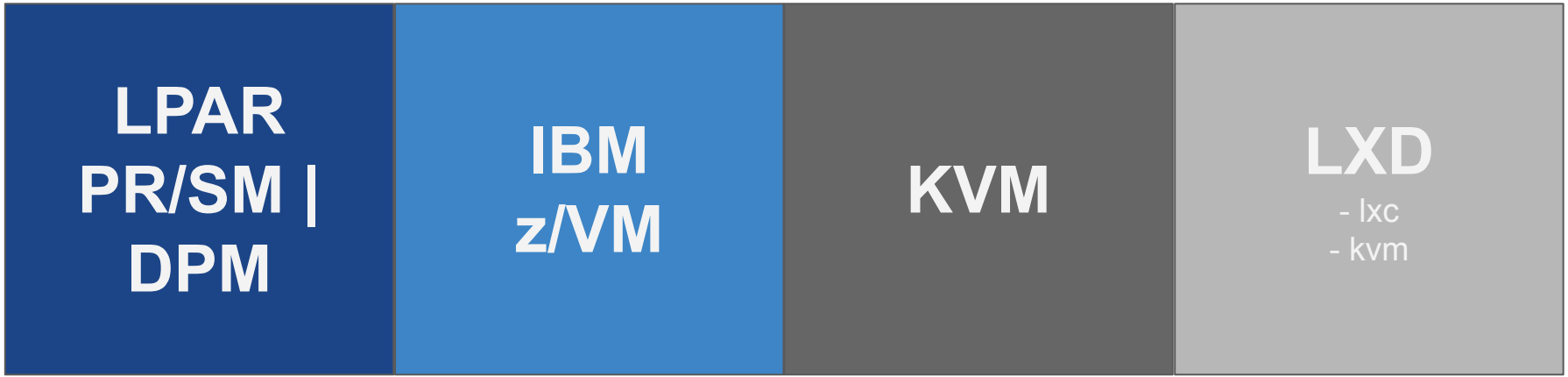
- MySQL
- MariaDB
- OpenStack
- RabbitMQ
- Wordpress
- HaProxy
- MemCache
- Kubernetes …

juju

https://jujucharms.com/q/?tags=ibm
https://jaas.ai/u/ibmcharmers



IBM Z and LinuxONE Hypervisors



IBM Z & LinuxONE - Juju Deployment Options

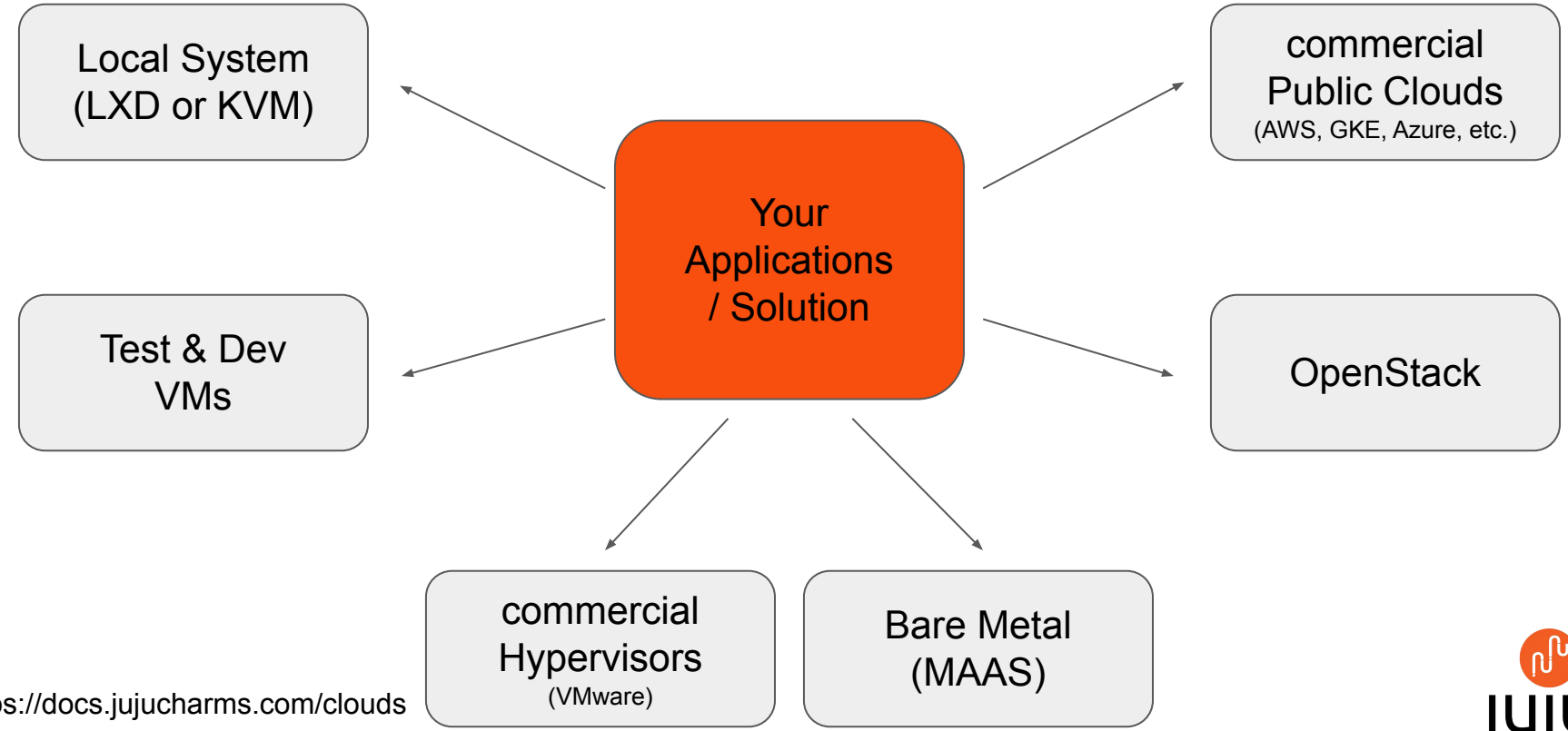
- ‘manual cloud ’ = list of pre-installed hosts
  - LPARs, z/VM guests, KVM VMs, LXD containers, …

- ‘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’ (tbd) or KVM (former MAAS Pods) *
  - Cloud experience cross pre-defined Ubuntu hosts
  - integration with other platforms managed by MAAS

* MAAS KVM support for s390x available since 2.5.3, recommended is using 2.6.x

Juju - Allows Reuse Across *Clouds*



https://docs.jujucharms.com/clouds

juju

Canonical Distribution of OpenStack (CDO)

openstack

Management & Automation

Infrastructure Services

Landscape & Autopilot

Juju

Nova

Horizon

Ceilometer /Telemetry

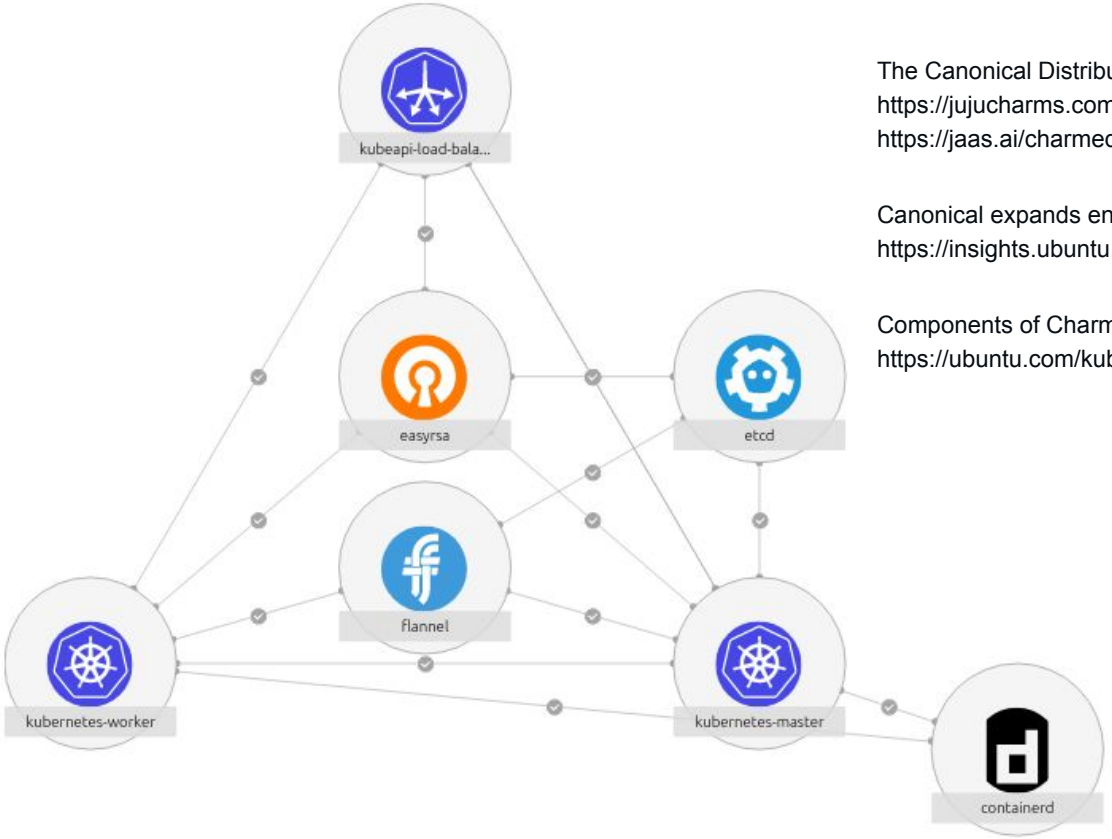
Keystone

Neutron

Ubuntu Server 16.04 LTS / 18.04 LTS / 20.04 LTS

MAAS
(on s390x KVM only)

CDK - Charmed Distribution of Kubernetes



The Canonical Distribution Of Kubernetes:
https://jujucharms.com/canonical-kubernetes/
https://jaas.ai/charmed-kubernetes

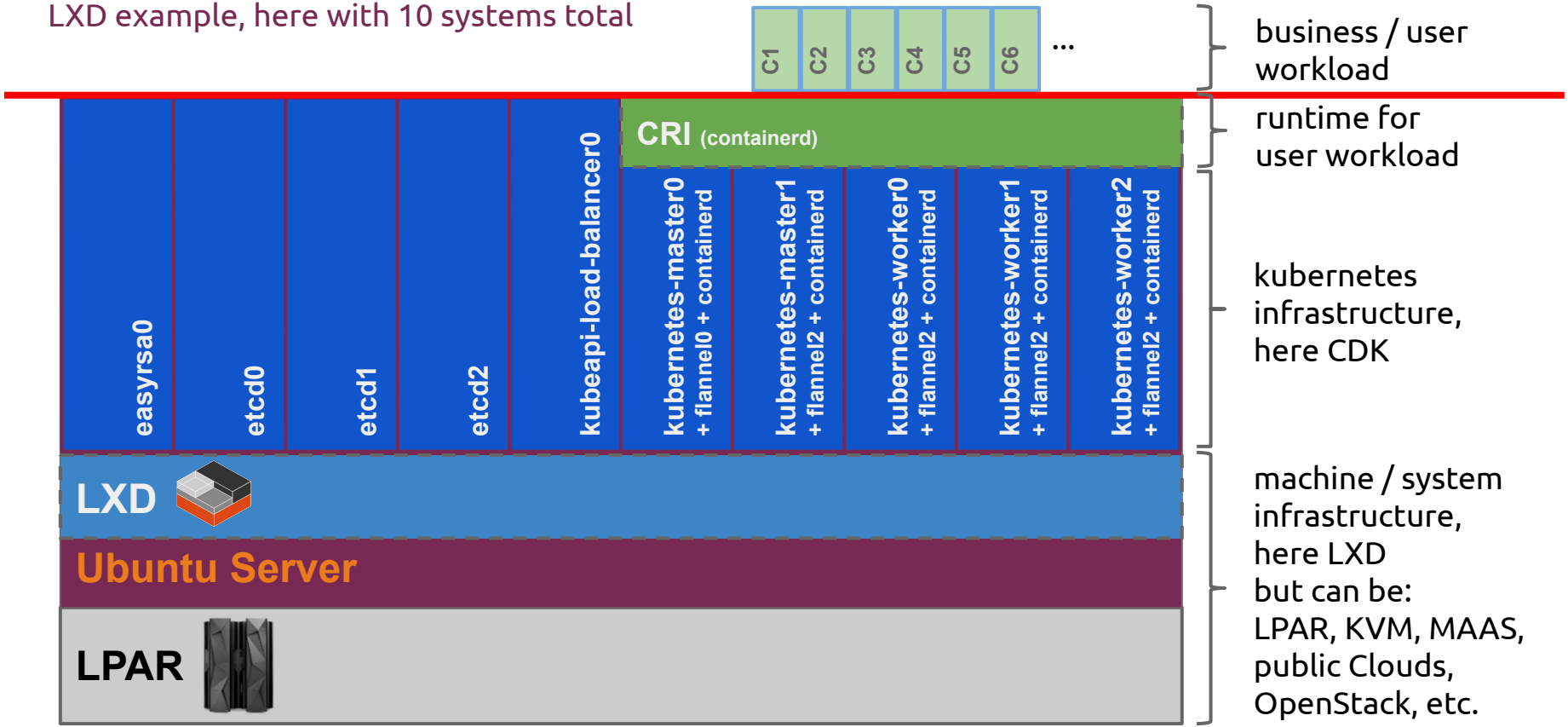
Canonical expands enterprise container portfolio:
https://insights.ubuntu.com/2016/09/27/canonical-expands-enterprise-container-portfolio/

Components of Charmed Kubernetes 1.20
https://ubuntu.com/kubernetes/docs/1.20/components

kubernetes

CDK Infrastructure

LXD example, here with 10 systems total

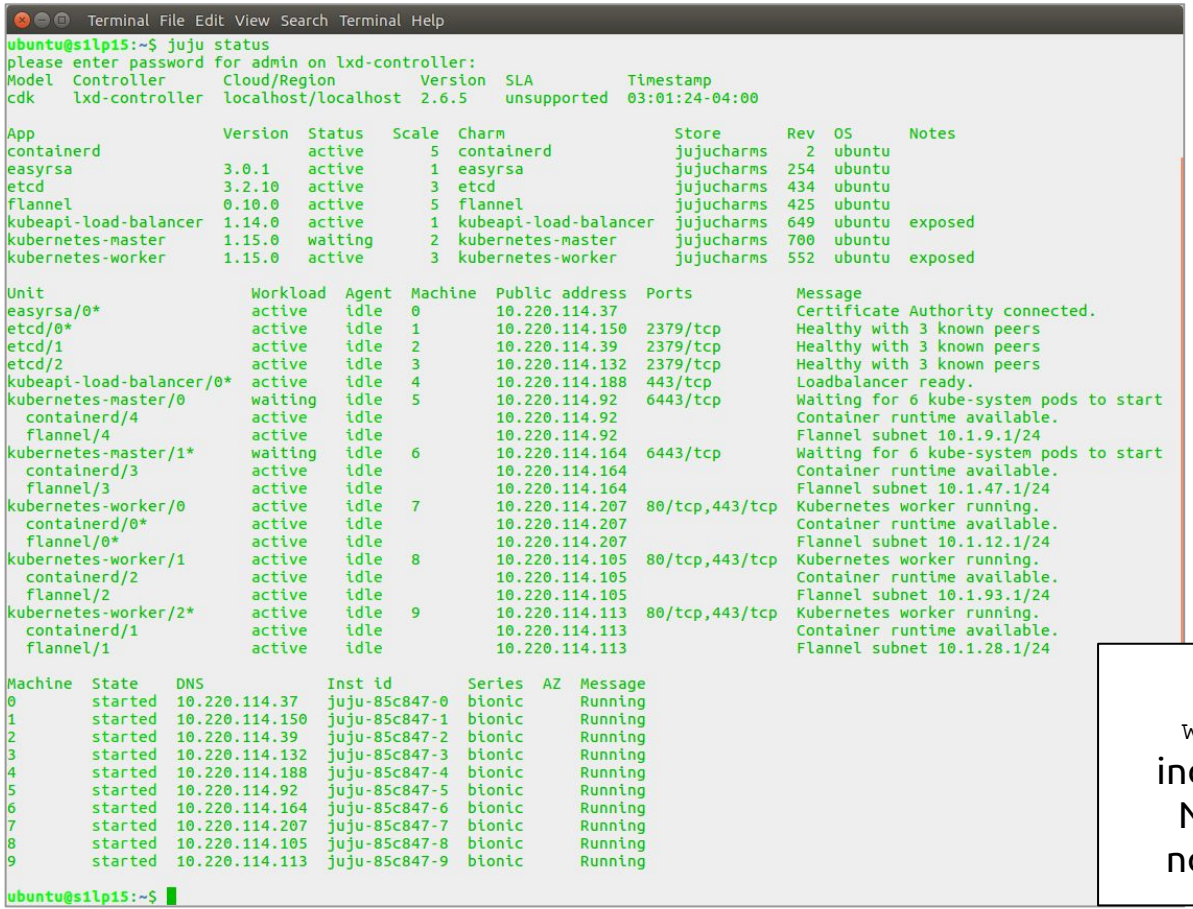


Simply try Kubernetes/CDK on IBM Z

Standard CDK environment with 10 systems using LXD local provider (but CDK can be stripped down ...)
LPAR resource requirement: 32GB RAM, 4 (shared) processors running Ubuntu Server 18.04

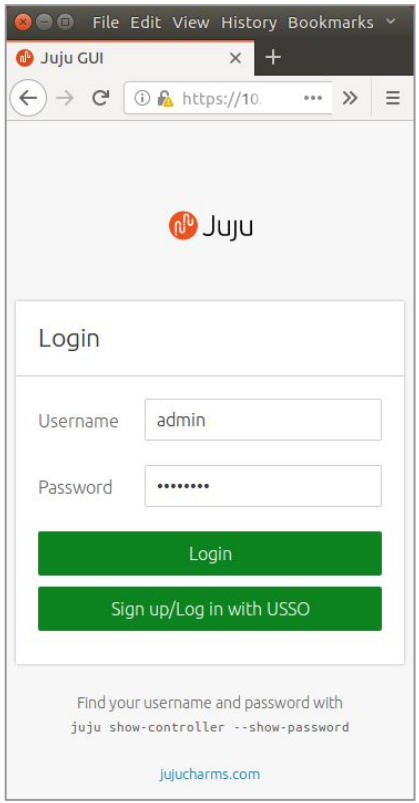
A
```
$ sudo apt -y -q update && sudo apt -y -q full-upgrade
$ sudo apt -y -q purge liblxc1 lxcfs lxd lxd-client
$ sudo apt -y -q install snapd
$ snap install lxd
$ lxd init                                  # use 'dir' storage backend and 'none' ipv6
$ snap install juju --classic
```
B
```
$ juju bootstrap localhost lxd-controller
$ juju add-model cdk
$ lxc profile edit "juju-cdk"               # adjust LXD profile
$ juju deploy canonical-kubernetes
$ watch -c juju status --color              # now watch and wait…
```
C
```
$ juju gui
GUI 2.14.0 for model "admin/cdk" is enabled at:
  https://10.0.8.11:17070/gui/u/admin/cdk
Your login credential is:
  username: admin
  password: d382qvf8vAPECAWEWCWÜC0JF0994    # OR: <unknown> (password changed by user)
$ sudo apt -y -q install net-tools
<workstation> $ sshuttle -r <user>@<remote_server> 127.0.0.1 <lxd network>
<workstation> $ firefox https://10.0.8.11:17070/gui/u/admin/cdk &
```

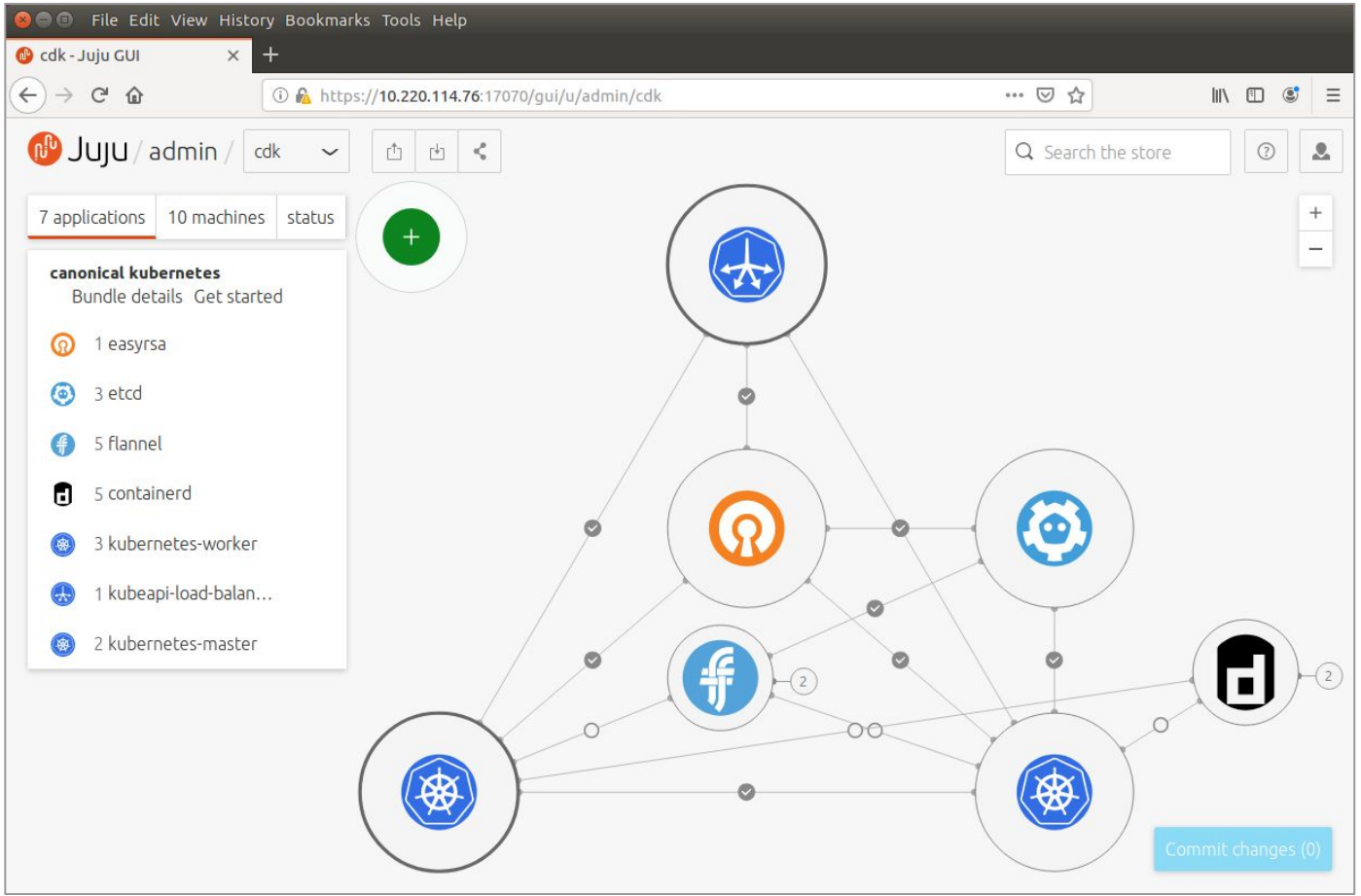
Kubernetes/CDK - juju status (cli)



`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/CDK Juju GUI - applications





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

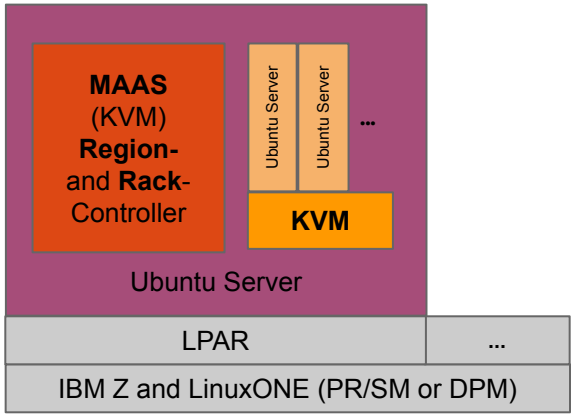
Cost effective at scale

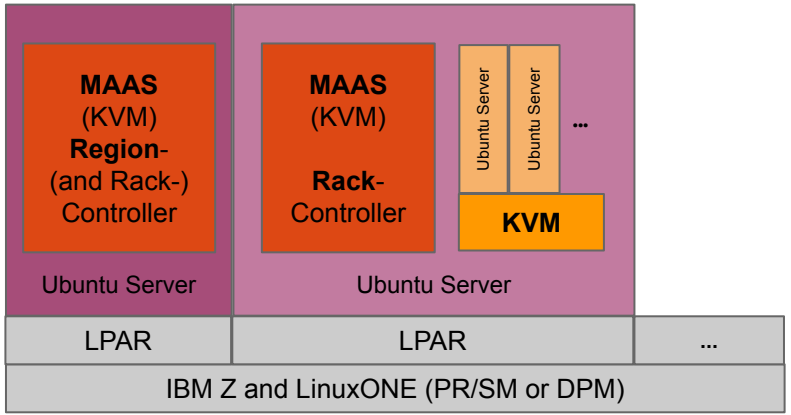
Bare metal operations with MAAS (tbd, today KVM only on s390x)

1. Manage your environment
2. Discover & manage your network
3. Manage your resources
4. Configure your hardware
5. Install your operating system

MAAS KVM (Pods) - on s390x

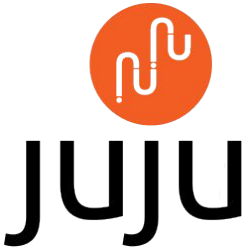
- MAAS is a solution for automated provisioning and dynamic re-purposing of (bare metal or) KVM VMs
- MAAS KVM (former MAAS Pods) is the part that allows to provisioning KVM VMs
- In addition MAAS provides some level of network management as well as manageability via APIs.
- The API is essential for further exploitation of provisioned machines by Juju.
- Initial availability with MAAS 2.5.3, recommended is 2.6.x or higher (ideally latest stable)
- The supported host operating system for MAAS (on s390x) is Ubuntu Server 18.04,
  but MAAS itself can run on other platforms/architectures, too.
- KVM can (but does not need to) run on the same system (LPAR) than MAAS.
- The deployed KVM VMs (guests) can be Ubuntu Server 20.04 LTS, 18.04 LTS and 16.04 LTS.



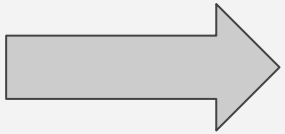


ubuntu
Supported by Canonical

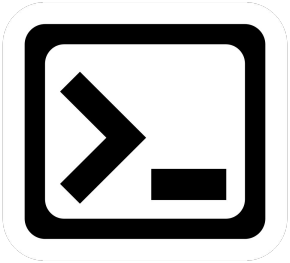
The **API** of MAAS
Is what provides the
most value.



Service
orchestration



Machine
configuration



Manual setup

3 Complementary Automation Tools

PHYSICAL and KVM PROVISIONING

+

DYNAMIC RE-PURPOSING

SERVICES MODELING, DEPLOYMENT

+

SCALING

ADMINISTRATION

+

AUDIT

+

COMPLIANCE

ubuntu
Supported by Canonical

* s390x: MAAS KVM only

More than you ever wanted to know about:

Hardware cryptographic support for IBM Z and LinuxONE with Ubuntu Server (70+ pages)

Hardware cryptographic support
for IBM Z and IBM LinuxONE
with Ubuntu Server

Klaus Bergmann, Reinhard Buendgen, Uwe Denneler, Jonathan Furminger,
Frank Heimes, Manfred Gnirss, Christian Rund, Patrick Steuer, Arwed Tschoeke

August 2, 2017

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

Ubuntu Server

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)

Install the **packages** needed for the hardware crypto support:

```
sudo apt-get install libica-utils libica? openssl-ibmca
```

Optional: Create a backup of the default openssl cofiguration file:

```
sudo cp -p /etc/ssl/openssl.cnf{,_$(date +%Y-%m-%d_%H:%M:%S).backup}
```

Append the **ibmca** related **configuration** lines to the OpenSSL configuration file (one line):

```
sudo tee -a ibmca section /etc/ssl/openssl.cnf <
/usr/share/doc/openssl-ibmca/examples/openssl.cnf.sample
```

Make sure that only **one "openssl_conf = openssl_def"** configuration **entry** exists in the config file at line 10, hence comment out any potential entries and insert an active entry at line 10:

```
sudo sed -i 's/^\(openssl conf = openssl def.*$\)/# \1/g' /etc/ssl/openssl.cnf
sudo sed -i '10i openssl_conf = openssl_def' /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 + CEX7S + Ubuntu Server 20.04**

Cryptographic algorithm support

function	hardware: dynamic	hardware: static	software
SHA-1	no	yes	yes
SHA-224	no	yes	yes
SHA-256	no	yes	yes
SHA-384	no	yes	yes
SHA-512	no	yes	yes
SHA-512/224	no	yes	yes
SHA-512/256	no	yes	yes
SHA3-224	no	yes	no
SHA3-256	no	yes	no
SHA3-384	no	yes	no
SHA3-512	no	yes	no
SHAKE-128	no	yes	no
SHAKE-256	no	yes	no
GHASH	no	yes	no
P_RNG	no	yes	yes
DRBG-SHA-512	no	yes	yes
ECDH	yes	yes	no
ECDSA Sign	yes	yes	no
ECDSA Verify	yes	yes	no
ECKGEN	yes	yes	no
Ed25519 Keygen	no	yes	no
Ed25519 Sign	no	yes	no
Ed25519 Verify	no	yes	no
Ed448 Keygen	no	yes	no
Ed448 Sign	no	yes	no
Ed448 Verify	no	yes	no
X25519 Keygen	no	yes	no
X25519 Derive	no	yes	no
X448 Keygen	no	yes	no
X448 Derive	no	yes	no
RSA ME	yes	no	no
RSA CRT	yes	no	no
DES ECB	no	yes	yes
DES CBC	no	yes	yes
DES OFB	no	yes	no
DES CFB	no	yes	no
DES CTR	no	yes	no
DES CMAC	no	yes	no
3DES ECB	no	yes	yes
3DES CBC	no	yes	yes
3DES OFB	no	yes	no
3DES CFB	no	yes	no
3DES CTR	no	yes	no
3DES CMAC	no	yes	no
AES ECB	no	yes	yes
AES CBC	no	yes	yes
AES OFB	no	yes	no
AES CFB	no	yes	no
AES CTR	no	yes	no
AES CMAC	no	yes	no
AES XTS	no	yes	no
AES GCM	no	yes	no

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 popluar and well documented:
  - Pervasive Encryption for Data Volumes: HTML or PDF
- It is straight forward, 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 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

all-in with Ubuntu 18.04 or higher

Ubuntu Server

Secure Boot (aka Secure IPL)

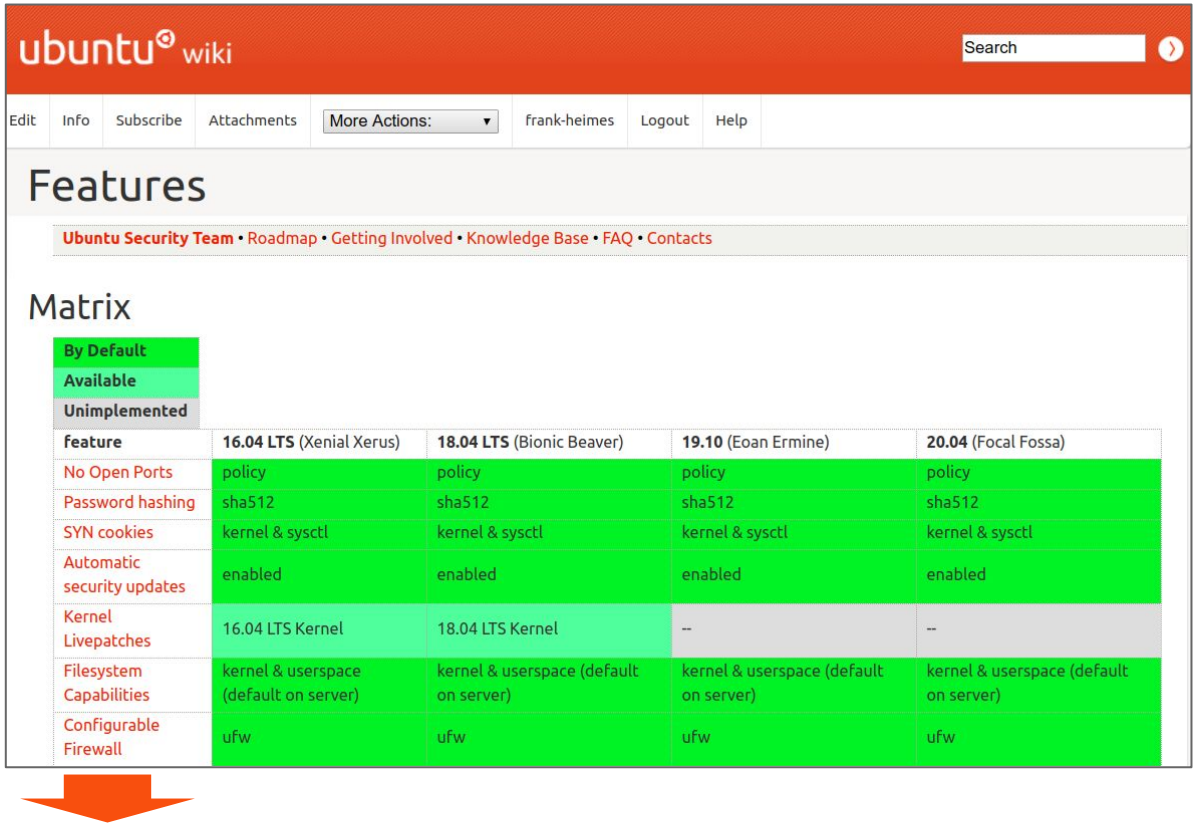
- Secure boot (for SCSI IPL) attributes to the Pervasive Encryption effort
- The IBM z15 and LinuxONE III hardware introduce secure boot (for SCSI IPL): requires a Kernel 5.3+ and s390-tools 2.9 (rec. 2.11) → Ubuntu 20.04 LTS
- HMC's Load task of the HMC now has a new check-box: 'Enable Secure Boot for Linux' in case 'SCSI Load' is selected.
- On 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 got incl. in the new Ubuntu live installer (subiquity).

Secure Execution (aka protected virtualization)

- 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 is: 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



ubuntu wiki

Search

Edit | Info | Subscribe | Attachments | More Actions: | frank-heimes | Logout | Help

Features

Ubuntu Security Team • Roadmap • Getting Involved • Knowledge Base • FAQ • Contacts

Matrix

By Default
Available
Unimplemented

feature	16.04 LTS (Xenial Xerus)	18.04 LTS (Bionic Beaver)	19.10 (Eoan Ermine)	20.04 (Focal Fossa)
No Open Ports	policy	policy	policy	policy
Password hashing	sha512	sha512	sha512	sha512
SYN cookies	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl
Automatic security updates	enabled	enabled	enabled	enabled
Kernel Livepatches	16.04 LTS Kernel	18.04 LTS Kernel	--	--
Filesystem Capabilities	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)
Configurable Firewall	ufw	ufw	ufw	ufw

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://wiki.ubuntu.com/Security/Features

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

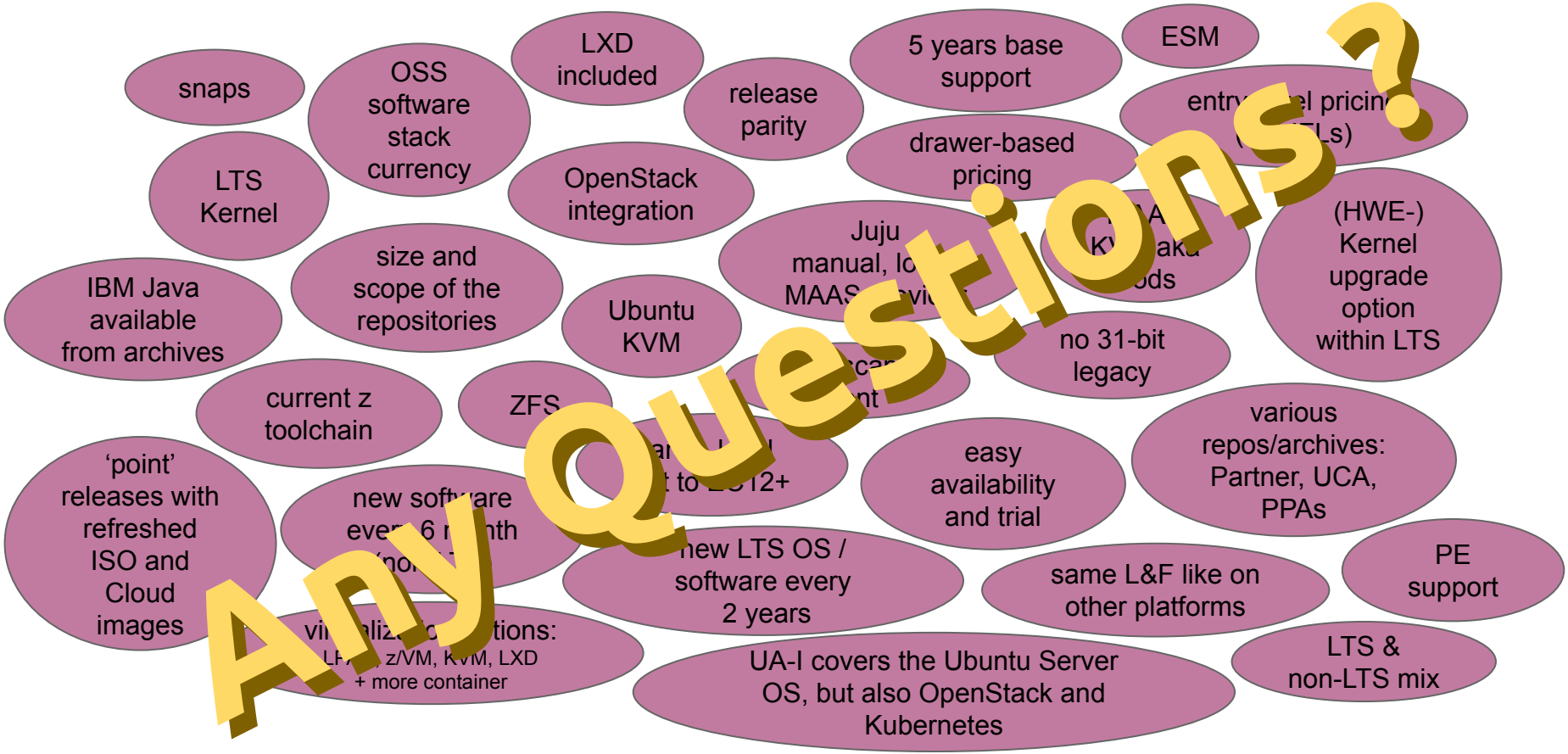
ESM - Extended Security Maintenance

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

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
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.

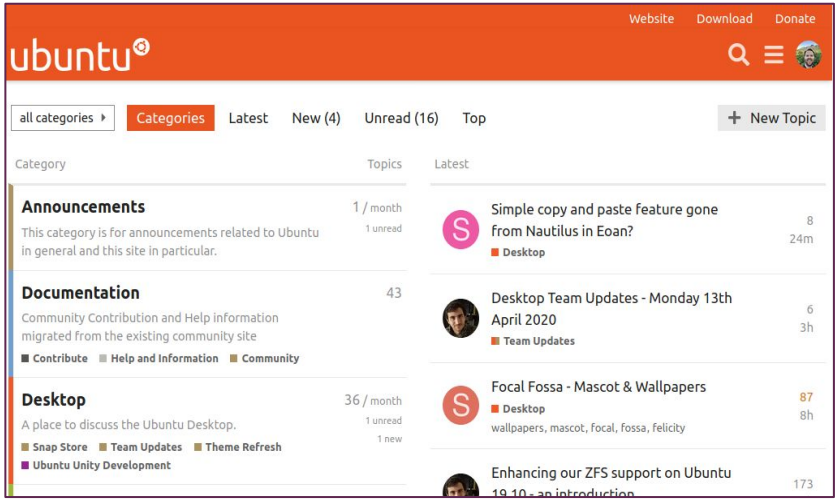
Why Ubuntu Server LTS on s390x?



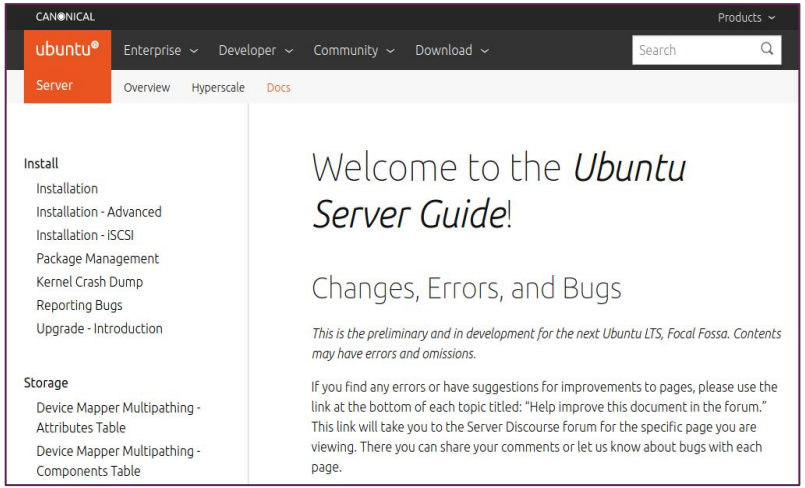
Thank you - Questions?

Thanks a lot - and stop by at:
https://ubuntu-on-big-iron.blogspot.com

Documentation and getting involved



https://discourse.ubuntu.com/



https://ubuntu.com/server/docs

Cloud Native & Confidential Computing on IBM Z & LinuxONE with Ubuntu 20.04 (webinar)

https://www.brighttalk.com/webcast/6793/440529/cloud-native-confidential-computing-on-ibm-z-linuxone-with-ubuntu-20-04