Ubuntu Server for IBM Z and LinuxONE

What’s New - November 2019

Frank Heimes, Tech. Lead Z, Canonical Ltd.

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

We are the company behind Ubuntu.

Delivered by Canonical
Ubuntu Server for IBM Z and LinuxONE (s390x)

Design Philosophy

- 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
- Consequentially tap into new client bases
- Exploit new features and components faster - in two ways:
  - hardware: zEC12/zBC12 and newer
  - software: latest kernels, compilers and optimized libraries
- Provide parity with other architectures
  - Release parity
  - Feature parity
  - Uniform user experience
  - Close potential gaps
- Open source - is collective power in action
- Upstream work and code only - no forks
- Offer a radically new pricing approach (drawer-based pricing) but also an entry-level pricing based on the number of IFLs (up to 4 IFLs)
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 is the 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 buildup to a release:

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.

Release Cadence - Ubuntu

https://wiki.ubuntu.com/Releases
https://wiki.ubuntu.com/LTS


Ubuntu 16.04 LTS

Ubuntu 18.04 LTS

Ubuntu 20.04 LTS

End-of-life

In service

In development

With s390x support

ESM

5 years
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](https://launchpad.net/ubuntu/bionic)
  Release date: April, 26th 2018
- Updated major components:
  - Kernel 4.15 (linux-generic)
  - 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](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 (1732423)
- support for new IBM hardware (1734748) (175437)
- AP bus kernel ABI for z14 (1734749)
- CPU-MF libpfm4 rebase for z13/z13s CPU-MF hardware counters (1741905)
- CPACF enhancements and acceleration for AES GCM (1735438) (1743750)
- HiperSocket enhancements and acceleration 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)

For an overall 18.04 release description, please see the official release notes: https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes
Ubuntu 18.10 (Cosmic Cuttlefish) reached End of Life on July 18 2019

Adam Conrad <adconrad@ubuntu.com>
to ubuntu-announce, ubuntu-security-announce

This is a follow-up to the End of Life warning sent earlier this month to confirm that as of today (July 18, 2019), Ubuntu 18.10 is no longer supported. No more package updates will be accepted to 18.10, and it will be archived to old-releases.ubuntu.com in the coming weeks.

The original End of Life warning follows, with upgrade instructions:

Ubuntu announced its 18.10 (Cosmic Cuttlefish) release almost 9 months ago, on October 18, 2018. As a non-LTS release, 18.10 has a 9-month support cycle and, as such, the support period is now nearing its end and Ubuntu 18.10 will reach end of life on Thursday, July 18th.

At that time, Ubuntu Security Notices will no longer include information or updated packages for Ubuntu 18.10.

The supported upgrade path from Ubuntu 18.10 is via Ubuntu 19.04. Instructions and caveats for the upgrade may be found at:

https://help.ubuntu.com/community/DiscoUpgrades

Ubuntu 19.04 continues to be actively supported with security updates and select high-impact bug fixes. Announcements of security updates for Ubuntu releases are sent to the ubuntu-security-announce mailing list, information about which may be found at:

https://lists.ubuntu.com/mailman/listinfo/ubuntu-security-announce

Since its launch in October 2004 Ubuntu has become one of the most highly regarded Linux distributions with millions of users in homes, schools, businesses and governments around the world. Ubuntu is Open Source software, costs nothing to download, and users are free to customise or alter their software in order to meet their needs.

On behalf of the Ubuntu Release Team,

Adam Conrad

ISO will move from:
http://cdimage.ubuntu.com/releases/18.10/release/to:
http://old-releases.ubuntu.com/releases/18.10/
Ubuntu 19.04 (Disco Dingo)

- The codename for the 19.04 release is 'DiscoDingo' or in short 'Disco':
  https://launchpad.net/ubuntu/disco
- Ubuntu Server non-LTS / development release
  Final Release: April, 18th 2019 (Release Candidate: April 11th 2019, Beta March 28th 2019)
- Major components:
  - kernel 5.0.0
  - qemu-kvm 3.1
  - libvirt 5.0
  - snapd 2.38
  - util-linux 2.33
  - gcc 8.3 (default), 9 (optional)
  - gdb 8.2
  - llvm 8.0
  - python 3.7.3 / 2.7.16
  - glibc 2.29
  - binutils 2.32
  - apt 1.8.0
  - openssl 1.1.1b
  - libica 3.4.0
  - opencryptoki 3.11
  - docker.io 18.09.5
  - netplan.io 0.96
  - smc-tools 1.2.0
  - s390-tools 2.8.0
Ubuntu 19.04 (Disco Dingo)

Non-complete list of s390x-specific new features and enhancements (since 18.10):

- Since s390x supports 1M huge pages (as well as 2GB huge pages, if requested) the support for libhugetlbfs (v2.19) was added for native s390x (LP:1823132) as well as for KVM (LP:1803315), so that customers running workloads with large memory footprints can benefit from improved memory performance.
- Now, having the kernel infrastructure (since 4.17) as well as the s390x-tool in place, the I/O device auto-configuration feature is ready to use. (LP:1776631)
- With (LP:1784643) Ubuntu Server is now prepared (with lib-zfcp-hbaapi) for port speed capabilities of 32GB line speed.
- The pass through capabilities for cryptographic resources got enhanced (LP:1787405).
- With the upgrade to qemu 3.1+ (LP:1798395) and the current kernel allows to make use of the latest KVM features for s390x.
- For example, but not limited, to PCI passthrough support for KVM (LP:1799446) and support for configurable virtio-crypto (LP:1802514).
- The enablement for virtio-gpu for s390x (LP:1799467) and (LP:1799472) now allows to better administer KVM virtual machines with UI based tools.
- The enablement for virtio-gpu for s390x (LP:1799467) and (LP:1799472) now allows to better administer KVM virtual machines with UI based tools.
- PCI Virtual function enablement and s390x-specific modifications were added as well (LP:1814684).
- Valgrind now has support for z13 (incl. SIMD) (LP:1799696).
- And a new PCI error reporting tool with support for NVMe was added to the s390-tools package (LP:1802499).
- Support for DPM's IO auto config feature introduced, that allows to simply enable hw resources in an LPAR 'like' you know from a PC.
- general support for DPM's IO auto config feature introduced, that allows to simply enable hw resources in an LPAR 'like' you know from a PC.
- hardware crypto pass-through for KVM,
- expanded SHA support and support for upcoming OSA Express* hardware
- further cryptsetup/zKey/PAES/LUKS2/PAM/EP11 and openencryptoki improvements
- optimistic usage of zkey in d-i for LUKS2 full disk encrypted installations
- The python-zhmcclient is now packaged and available via universe (LP:1805367).
- smc-tools are upgraded to 1.2.0, now including smc_rnics and smc_dbg (LP:1815425).
- The support for TSO (TCP Segmentation Offload) in ipv4 and ipv6 for Layer2 was extended (LP:1805793).
- efis-lockdown was fixed to restrict debugfs when the kernel is locked down (LP:1807686).
- Support to allow the protected key AES (paes) module to derive protected keys from clear keys (LP:1811354).
- DIF and DIF+DIX integrity protection mechanisms in FCP can now be separately configured (LP:1814537).
- Application can access new data sets that were created after zfs was mounted - without the need to remount zfs (LP:1814538).
- That said s390-tools were upgraded to 2.8.0 and support was added for automatic usage of zkey and pervasive encryption by the installer (LP:1766865).
- When using the installer and selecting "Guided - use entire disk and setup encrypted LVM" zkey will be used automatically, in case a CryptoExpress domain is available with a master key set.
- qeth performance (OSA and Hipersockets) got improved by general code changes (LP:1814899) and link speed enhancements were made in kernel (LP:1814891) and in the tooling (LP:1814892) - in preparation for future network hardware.
- New instruction support was added to binutils, leading to improved performance with that enhanced instruction support (LP:1815040).
- The new glibc version 2.29 (LP:1817523) has several minor improvements for s390x, and with that vx and vxe are now marked as important hwcap, to be able to provide differently tuned shared libraries (LP:1821200).

For an overall 19.04 release description, please see the official release notes: https://wiki.ubuntu.com/DiscoDingo/ReleaseNotes
Ubuntu 19.10 (Eoan Ermine)

- The codename for 19.10 is 'Eoan Ermine' or just 'Eoan': [https://launchpad.net/ubuntu/eoan](https://launchpad.net/ubuntu/eoan)
- Ubuntu Server non-LTS / development release [announcement](https://launchpad.net/ubuntu/eoan)
- Major components:
  - kernel 5.3
  - qemu-kvm 4.0
  - libvirt 5.4
  - snapd 2.41
  - util-linux 2.34
  - gcc 9.2
  - gdb 8.3
  - python 3.7.5 / (2.7 universe)
  - golang 1.12
  - valgrind 3.15
  - OpenJDK 11 (default) (8 & 13, 14 in archive)
  - glibc 2.30
  - binutils 2.33
  - apt 1.9.4
  - openssl 1.1.1c
  - openssl-ibmca 2.1.0
  - libica 3.6.0
  - opencryptoki 3.11.1
  - docker.io 19.03.2
  - netplan 1.10
  - libhugetlbfs v2.21
  - smc-tools 1.2.1
  - s390-tools >= 2.11.0
Ubuntu 19.10 (Eoan Ermine)

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

- Enhanced support for existing and especially new hardware (LP:1830742), (LP:1842774) is a primary task, not only with kernel focus - new hardware CPU models got introduced (LP:1830239), enhanced hardware diagnose added (LP:1829270), the diagnose data for the Linux kernel enhanced (LP:1829270), but also with virtualization/KVM focus - with IO enhancements introduced for KVM guests (LP:1834533), DASD passthrough support added (LP:1843961) and again the new hardware models for Qemu/KVM (LP:1830238), (LP:1836066) and (LP:1828038).
- Based on the new IBM z15 and LinuxONE III hardware generation secure boot (IPL) was introduced for zFCP/SCSI disks. This affects not only the kernel (LP:1829027), (LP:1830617), (LP:1843960), (LP:1843961), but also the installer and QEMU/KVM (LP:1830243).
- The s390-tools upgrade to v2.11.0 comes with several enhancements, not limited to secure boot (IPL) support for SCSI (LP:1825351), (LP:1843879) as well as zkey improvements (LP:1836907).
- The CPU_MF hardware counters were significantly enhanced as well - this work concerned again several components like the kernel (LP:1834201), (LP:1836739), (LP:1836340), but also perf (LP:1837051) and libpfm (LP:1837016).
- In addition virt-manager was updated (LP:1827069) as well as boot configuration (LP:1826856).
- There is now also improved hardware support for zlib to gain performance improvements (LP:1823157).
- Patches got included for optimized s390x zlib compression (LP:1825350) and to increase gzip performance (LP:1839123), (LP:1841052).
- The kernel compression method was changed to LZ4 to improve boot speed (LP:1840934), (LP:1841193).
- Version bump to new upstream release of libhugetlbfs v2.21 (LP:1825216).
- smc-tools got upgraded to latest v1.2.1 (LP:1825217).
- Upgrade to latest upstream valgrind v3.15 with some improvements and s390x fixes (LP:1828219).
- Updated toolchain to latest gcc 9.2 (LP:1825346) and LLVM 9.0 (LP:1836343).
- And with lifting glibc to 2.30 new instructions support were added that leads to performance improvements (LP:1825349).
- The upgrade of libdfp to v1.0.14 introduces significant s390x decimal floating-point hardware improvements and exploitations (LP:1836532).
- libaltas libraries are not available in optimized versions for selected s390x hardware - z13 and z14 (LP:1837577).
- The PCI subsystem obtained some modifications and fixes for MIO (LP:1825352), (LP:1844668) and directed interrupt support (LP:1825353).
- SIMD accelerated implementations for poly1305 (LP:1736704) and chacha20 (LP:1736705) were added to openssl.
- On top openssl-ibmca was lifted to the very latest v2.1.0 (LP:1826198), (LP:1836865) as well as libica(3) to v3.6.0 (LP:1826194), (LP:1836866).
- With the upgrade of opencryptoki to v3.11.1 (LP:1826193), that includes various bug fixes, support for the opencryptoki ioca tokens CKM_SHA256_RSA_PKCS_PSS, CKM_SHA384_RSA_PKCS_PSS and CKM_SHA512_RSA_PKCS_PSS was added (LP:1835048) - especially in regard to TLS 1.3 and its GSKit support.
- Further cryptography improvements include libp11 upgrade to v0.4.10 (LP:1830730) in support of OpenSC, zcryptstats added to the s390-tools package to introduce support for measurements of crypto hardware (LP:1835554) and kernel crypto modification on the seeding of PRNG with (on-chip CPACF-based) TRNG (LP:1835553) and the concurrent handling (add and remove events) of AP facilities LUNs without the need of the (partly considered unstable) DIX feature.
- Finaly several security and integrity aspects got addressed, for example with the introduction of KASLR (kernel address space layout randomization) - to support the load of the kernel to a random location in protection against certain attacks (LP:1832626) and the splitting of DIF and DIX boot time controls (LP:1836608) that now allows hardware-based DIF data integrity checking between FCP channel and target LUNs without the need of the (partly considered unstable) DIX feature.

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

|-------|-------|-------|-------|----------|-------|-------|-------|----------|-------|-------|

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

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](https://wiki.ubuntu.com/Releases)

Even if the initial plan was to release just 5 point releases for an Ubuntu LTS version, a 6th point release was made available for 16.04 - so there is indeed a 16.04.6 available!

16.04.6 is a security-targeted release for the purpose of providing updated installation media which protects new installations from the recently discovered APT vulnerability (USN-3863-1).

Many other security updates for additional high-impact bugs are of course included as well, with a focus on maintaining stability and compatibility with Ubuntu 16.04 LTS.

The life-time of 16.04 does not change with this additional point release, nor the kernel (except for some fixes).

The 16.04.6 ISO image still offers the choice of either booting the installer with the default 16.04 GA Kernel 4.4 or with the HWE-Kernel, similar to the 18.04 GA kernel 4.15 (both on the very latest patch level).

In other words the following two installer kernels are available:

```
/boot   -   Kernel 4.4.0-142 (default)
/boot-HWE   -   Kernel 4.15.0-45 (from 18.04 / Bionic)
```

The default Kernel is usually still kernel 4.4 - until you explicitly change it and 'opt-in' to use the HWE-Kernels.

Maintenance updates are still provided for 5 years (in total) for Ubuntu Server and the Ubuntu Server Cloud image.

Users of the Ubuntu Cloud image are recommended to launch new instances using the latest images from the 16.04 release stream, which include the fix in APT for USN-3863-1.

The updated 16.04.6 Cloud image can be found here. https://cloud-images.ubuntu.com/releases/16.04/release/

The general download links for "Ubuntu 16.04.6 LTS (Xenial Xerus)" are:

https://www.ubuntu.com/download/server/s390x
https://www.ubuntu.com/download/server/linuxone
This is a distilled view of the 18.04.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 18.04.2 there is a choice of:
- 2 installation kernels (for ‘d-i’)
- 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
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, LinuxONE Emperor / Rockhopper
  and LinuxONE Emperor II / Rockhopper II and even on zPDT

- in classic or DPM (Dynamic Partition Manager) mode

* The Charmed Distribution Of Kubernetes: https://jaas.ai/canonical-kubernetes

Ubuntu Server Certified Hardware (s390x)

**zBC12/zEC12**
- IBM zBC12 (LPAR) Server
- IBM zBC12 (z/VM) Server
- IBM zEC12 (LPAR) Server
- IBM zEC12 (z/VM) Server

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

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

**z15 T01 Max34-Max190**
- IBM z15 (LPAR DPM & classic) Server
- IBM z15 (z/VM) Server

**LinuxONE**
- IBM LinuxONE Emperor I+II (LPAR DPM) Server
- IBM LinuxONE Emperor I+II (LPAR classic) Server
- IBM LinuxONE Emperor I+II (z/VM) Server
- IBM LinuxONE Rockhopper I+II (LPAR DPM) Server
- IBM LinuxONE Rockhopper I+II (LPAR classic) Server
- IBM LinuxONE Rockhopper I+II (z/VM) Server
- IBM LinuxONE III (LPAR DPM & classic) Server
- IBM LinuxONE III (z/VM) Server

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
IBM Z and LinuxONE - Tested platforms

IBM tested and Partner certified Linux environments
Learn about the IBM tested and the certified Linux environments of the distribution partners. Check the statements of the individual Linux distribution 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.

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
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
- **CDO** - Canonical’s Distribution of OpenStack with Nova KVM support

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

Check out what **LXD** can do for you, too!
Ubuntu KVM

- Available for all architectures supported by Ubuntu Server - including s390x.
- **Upstream** KVM (qemu-kvm) and libvirt (libvirt-bin)
- Exploits hardware assisted virtualization of the s390x architecture (SIE instruction).
- Supported on all IBM Z and LinuxONE hardware that’s supported by Ubuntu Server (>= EC12).
- Classic PR/SM (Processor Resource and System Management) or DPM (Dynamic Partition Manager).
- Take, install and use it from the **standard Ubuntu archive** (main).
- **Easy installation** - just do: `sudo apt install qemu-kvm libvirt-bin`
- **Integral** part of **Ubuntu Server** - no additional subscription on top of UA needed.
- By default **secured** with `apparmor`.
- Ubuntu KVM is part of the hardware **certification**.
- **Nested virtualization** (KVM on top of z/VM) is **not** supported.
- VMs can be setup and used in many flexible ways:
  - kvm aka qemu-system-s390x -enable-kvm (low-level cmd-line)
  - virsh (libvirt command-line)
  - virt-manager (graphical, connect to qemu-kvm from remote)
  - virt-install (programs to create and clone virtual machines)
  - Canonical / Ubuntu OpenStack (using Nova-KVM)
  - uv-tool and simplestreams (library and tools for using Ubuntu Cloud Images with libvirt)
- **Networking options are**: bridge, macvtap and openvswitch.
- Migration document - IBM KVM to Ubuntu KVM: [https://wiki.ubuntu.com/IBMKVMtoUbuntuKVM](https://wiki.ubuntu.com/IBMKVMtoUbuntuKVM)

---

**Check out what LXD can do for you, too!**
uv-tool (an Ubuntu hidden treasure ;-)

Ubuntu virtualisation front-end for libvirt and KVM

user@workstation:~$ ssh ubuntu@server
server:~$ ssh-keygen -t rsa
...
server:~$ sudo apt -y -q install uvtool-libvirt
...
# you may need to re-login!

server:~$ NAME=bionic
arch=$arch label=release release=$NAME
server:~$ uvt-kvm create $NAME-vm release=$NAME label=release
server:~$ uvt-kvm wait $NAME-vm
server:~$ uvt-kvm ssh $NAME-vm
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-48-generic s390x)
...
ubuntu@bionic-vm:~$ exit
logout
Connection to 192.168.123.45 closed.

server:~$ uvt-kvm ssh bionic-vm
...
uv-tool (an Ubuntu hidden treasure ;-)

One-Liner example for 18.04 / bionic - already released:
$ NAME=bionic; uvt-simplestreams-libvirt sync --source http://cloud-images.ubuntu.com/releases arch=$(arch) label=release release=$NAME && 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:~$

One-Liner example for 19.10 / eoan - currently in development / daily:
$ NAME=eoan; uvt-simplestreams-libvirt sync --source http://cloud-images.ubuntu.com/daily arch=$(arch) label=daily release=$NAME && uvt-kvm create $NAME-vm release=$NAME label=daily && uvt-kvm wait $NAME-vm && uvt-kvm ssh $NAME-vm

Notice that ‘daily’ images are not produced for all Ubuntu releases, just for those that are in active development.

LXD is a container ‘hypervisor’ and a new user experience for lxc. It builds on top of lxc and is no replacement of lxc.

LXD is available for all architectures supported by Ubuntu Server - including s390x.

LXD is a daemon exporting an authenticated representational state transfer application programming interface (REST API) both locally over a unix socket and over the network using https.

LXD is modern hypervisor for *machine* containers (in contrast to application containers, like Docker) and provides:

- Two clients for the LXD daemon, one is an OpenStack plugin (Nova-LXD),
- the other a standalone command line tool – but use the REST interface
- Security by default (unprivileged containers, apparmor, seccomp, etc.)
- Image based workflow (no more locally built rootfs)
- Support for online snapshots, incl. running state, rollbacks and clones
- Live migration support (requires CRIU, available for s390x since 17.10, experimental)
- Shell command control

https://linuxcontainers.org/lxd/
The biggest new feature for LXD 3.0 is the introduction of clustering support.

VM filesystem sync - lxd-p2c makes it possible to import a system’s filesystem into a LXD container using the LXD API.

Support for NVIDIA runtime passthrough

Newly introduced nvidia.runtime container configuration key, combined with a copy of the nvidia-container-cli tool and liblxc 3.0 now makes it possible to automatically detect all the right bits on the host system and pass them into the container at boot time.

Clustering:

The biggest new feature for LXD 3.0 is the introduction of clustering support.

This allows for identically configured LXD servers to be joined together as part of a cluster, appearing to the outside world as one big LXD server.

The LXD database is replicated using dqlite (a combination of sqlite3 and raft), making it so that 3 of the cluster members have a copy of the entire database at any given time.

No special system configuration or services are required to setup LXD clustering, all you need is a few available machines or VMs with similar network and storage properties, then lxd init will walk you through the process of creating the cluster and then joining some servers into it.
IBM Z and LinuxONE Hypervisors

- LPAR
- PR/SM
- IBM z/VM
- KVM
- LXD
  ‘container hypervisor’
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 - The Services Modeling Tool
Brings all our Open Source Packages and IBM Software

https://jujucharms.com/q/?tags=ibm
https://jaas.ai/u/ibmcharmers
https://jaas.ai/openstack-base/bundle/60

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

Browse the store >
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 support through KVM (former MAAS Pods) *
  - Cloud experience cross pre-installed Ubuntu KVM 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

Local System (LXD or KVM)

Test & Dev VMs

Your Applications / Solution

commercial Public Clouds (AWS, GKE, Azure, etc.)

OpenStack

commercial Hypervisors (VMware)

Bare Metal (MAAS)

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 16.04 LTS / 18.04 LTS

MAAS
(on s390x KVM only)

intel
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:
CDK - Charmed Distribution of Kubernetes

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

Introducing Kubernetes version 1.0!: https://insights.ubuntu.com/2015/07/21/introducing-kubernetes-version-1-0/ (s390x support for 1.8)


Kubernetes 1.15 now available from Canonical https://ubuntu.com/blog/kubernetes-1-15-now-available-from-canonical

Kubernetes 1.16 available from Canonical https://ubuntu.com/blog/kubernetes-1-16-available-from-canonical
Kubernetes High Level Architecture
CDK Infrastructure
LXD example, here with 10 systems total

- easyrsa0
- etc0
- etc1
- etc2
- kubeapi-load-balancer0
- etcd0
- etcd1
- etcd2
- kubernetes-master0 + flannel0 + containerd
- kubernetes-master1 + flannel1 + containerd
- kubernetes-worker0 + flannel2 + containerd
- kubernetes-worker1 + flannel2 + containerd
- kubernetes-worker2 + flannel2 + containerd
- kubelet

CRI (containerd)

- runtime for user workload
- kubernetes infrastructure, here CDK
- machine / system infrastructure, here LXD

but can be:
LPAR, KVM, MAAS, public Clouds, OpenStack, etc.

business / user workload
CDK Components

The Canonical Kubernetes (https://jujucharms.com/canonical-kubernetes/) bundle (without the redundant components) contain:

- 1 unit of **easyrsa**
  EasyRSA is a command line utility to build and manage Public Key Infrastructure (PKI) Certificate Authority (CA).
- 3 unit of **etcd**
  Etcd is a highly available distributed key value store.
- 1 unit of **kubeapi-load-balancer**
  A round robin Nginx load balancer to distribute traffic for Kubernetes apiservers.
- 2 unit of **kubernetes-master**
  The Master Node is responsible for managing the Kubernetes cluster.
- 3 unit of **kubernetes-worker**
  A Worker Node is a node which runs the applications, it is controlled by Kubernetes Master Node.
- 5 units of **flannel** (subordinate)
  Flannel is a generic overlay network, will be used as Container Network Interface (CNI) Network Provider for our Kubernetes cluster. Flannel will run on all kubernetes-master and kubernetes-worker nodes.
  Flannel plugs in to kubernetes Cluster Network Interface, CNI
- 5 units of **containerd** (subordinate)
  Manages the container lifecycle of its host system (worker and master) - from image transfer and storage to container execution and supervision to low-level storage to network attachments and beyond.
  Containerd plugs in to kubernetes Cluster Runtime Interface, CRI

Note: A juju bundle is a description of a deployment in terms of charms to use, their relations, number of units of each, unit placement, values of the parameters, etc.
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

```
$ 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
$ snap install juju --classic
$ 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...
$ 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: d382qvf8vAPECASEWOC0JF0994  # 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**.
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>IPV4</th>
<th>IPV6</th>
<th>TYPE</th>
<th>SNAPSHOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>juju-350b93-0</td>
<td>RUNNING</td>
<td>10.220.114.76 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-0</td>
<td>RUNNING</td>
<td>10.220.114.37 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-1</td>
<td>RUNNING</td>
<td>10.220.114.150 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-2</td>
<td>RUNNING</td>
<td>10.220.114.39 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-3</td>
<td>RUNNING</td>
<td>10.220.114.132 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-4</td>
<td>RUNNING</td>
<td>10.220.114.188 (eth0)</td>
<td></td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-5</td>
<td>RUNNING</td>
<td>10.220.114.92 (eth0)</td>
<td>10.1.9.0 (flannel.1)</td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-6</td>
<td>RUNNING</td>
<td>10.220.114.164 (eth0)</td>
<td>10.1.47.0 (flannel.1)</td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-7</td>
<td>RUNNING</td>
<td>10.220.114.207 (eth0)</td>
<td>10.1.12.0 (flannel.1)</td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-8</td>
<td>RUNNING</td>
<td>10.220.114.105 (eth0)</td>
<td>10.1.93.0 (flannel.1)</td>
<td>PERSISTENT</td>
<td></td>
</tr>
<tr>
<td>juju-85c847-9</td>
<td>RUNNING</td>
<td>10.220.114.113 (eth0)</td>
<td>10.1.28.0 (flannel.1)</td>
<td>PERSISTENT</td>
<td></td>
</tr>
</tbody>
</table>
Kubernetes/CDK Juju GUI - applications
Kubernetes/CDK Juju GUI - status
<table>
<thead>
<tr>
<th></th>
<th>Kubernetes Explorer</th>
<th>Kubernetes Discoverer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environments</strong></td>
<td>AWS, Azure, Google Cloud, Oracle Cloud, VMware, Canonical OpenStack</td>
<td>Add bare metal</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>5-200 nodes</td>
<td>5-2,000+ nodes</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>VMware, Cloud block, Local, NFS</td>
<td>Add Ceph, software-defined storage acceleration, custom integration of existing on-premises storage</td>
</tr>
<tr>
<td><strong>Networking</strong></td>
<td>Nodeport, Flannel</td>
<td>Add Calico, optional custom CNI SDN</td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td>SAML, OAuth2 via Dex, RBAC</td>
<td>Add Active Directory, LDAP</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>Web video conference. 2 x 4-hour sessions</td>
<td>On site, classroom style. 4 days in the same week</td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td>30 days 10x5 office hours phone support included. Optional ongoing enterprise support and operations from Canonical</td>
<td></td>
</tr>
<tr>
<td><strong>Upgrades</strong></td>
<td>Available within 7 days of upstream release</td>
<td></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Security patches for entire stack, from kernel to kubernetes. All CVEs and additional security improvements included</td>
<td></td>
</tr>
<tr>
<td><strong>Connectivity</strong></td>
<td>Internet access required</td>
<td>Offline deployment possible</td>
</tr>
<tr>
<td><strong>QOS</strong></td>
<td>n/a</td>
<td>CPU pinning, resource limits and quotas</td>
</tr>
<tr>
<td><strong>Monitoring</strong></td>
<td>Prometheus / ELK</td>
<td>Custom monitoring integration possible</td>
</tr>
</tbody>
</table>
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 (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.
- 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 19.04, 18.10, 18.04 LTS and 16.04 LTS.

<table>
<thead>
<tr>
<th>MAAS (KVM)</th>
<th>Region- and Rack- Controller</th>
<th>KVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu Server</td>
<td>Ubuntu Server</td>
<td>...</td>
</tr>
<tr>
<td>LPAR</td>
<td>...</td>
<td>IBM Z and LinuxONE (PR/SM or DPM)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAAS (KVM)</th>
<th>Region- (and Rack-) Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu Server</td>
<td>...</td>
</tr>
<tr>
<td>LPAR</td>
<td>...</td>
</tr>
<tr>
<td>IBM Z and LinuxONE (PR/SM or DPM)</td>
<td>...</td>
</tr>
</tbody>
</table>
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 features

MAAS KVM (Pods)

1. Create LPAR(s) and install Ubuntu server into LPAR(s) -- as usual

2. Configure KVM in Ubuntu host and register KVM in MAAS as a pod (type virsh)

3. KVM VMs are created and OS deployed into VMs using MAAS API
   more efficient use of resources by dynamically allocating hardware
The API of MAAS is what provides the most value.

Please bring up a machine with 24 GB RAM, a separate root partition and 2 additional partitions.
MAAS

KVM aka Pods
MAAS (virtual) Machine live-cycle

- Machines: 1 machine available
- Filters: in:(Selected)
- 1 Machine, 1 Resource pool
  - Check: humane-orca.maas
  - IP: 192.168.122.202 (PXE)
  - Status: 18.04 LTS
  - Owner, Tags: admin, virtual, pod-c...

MAAS name: s11p11 MAAS
MAAS version: 2.5.3 (7533-g65952b418-0ubuntu1-18.04.1)

View release notes - View documentation - Legal information - Give feedback

© 2019 Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd.
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.
Protecting Data at Rest

Optimistic Usage of zkey in Ubuntu Server 19.04 Installer or higher

- Protecting data at rest in the context of Pervasive Encryption is very popular and well documented here:
  - Pervasive Encryption for Data Volumes:
    - [http://public.dhe.ibm.com/software/dw/linux390/docu/l5n1dc03.pdf](http://public.dhe.ibm.com/software/dw/linux390/docu/l5n1dc03.pdf)
  - Setting up Data Volumes for Pervasive Encryption – In less than ten minutes (video):
    - [https://ibm.biz/BdzMs6](https://ibm.biz/BdzMs6) | [https://youtu.be/t2Ph_h0LcsQ](https://youtu.be/t2Ph_h0LcsQ)

- It's pretty straightforward, but requires manual steps to setup - even for supplemental (data) volumes. And it becomes pretty tricky in case the root filesystem and maybe the swap device should be encrypted in the same way! But here is where the **optimistic usage of zkey** of the **installer** (d-i) of **Ubuntu Server 19.04** (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 accessible to the installation system
    - and an initial master key configured (either with TKE or the 'IBM CCA Host Libraries and Tools')
    - and either an ECKD/DASD or an zFCP/SCSI disk
  - now perform a common d-i installation, and at the “**Partition disks**" screen just choose: “Guided - use entire disk and set up encrypted LVM”, and with that:
    - d-i will *try* to use zkey to obtain a secure key in XTS cipher mode
    - If d-i is able to obtain one (in case all the above requirements are in place)
      - d-i will use it for a LUKS2 full disk encrypted installation (for root and swap)
    - If not, d-i will fall-back to the traditional pass-phrase usage

- Notice that d-i will always ask for a pass-phrase, it's needed to unlock the meta data, even in case of zkey usage.
Protecting Data at Rest

How does it look like using optimistic zkey usage (in Ubuntu Server 19.04 and higher)

```
$ sudo cryptsetup status $(awk '{ print $1 }' /etc/crypttab)
/dev/mapper/mpatha5_crypt is active and is in use.
type: LUKS2
cipher: paes-xts-plain64
keysize: 1024 bits
key location: keyring
device: /dev/mapper/mpatha5_crypt
sector size: 4096
offset: 32768 sectors
size: 132683776 sectors
mode: read/write
flags: discards
```
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
Why Ubuntu Server LTS on s390x?

- snaps
- LTS Kernel
- IBM Java available from archives
- current z toolchain
- 'point' releases with refreshed ISO and Cloud images
- virtualization options: LPAR, z/VM, KVM, LXD + more container
- new software every 6 month (non-LTS)
- new LTS OS / software every 2 years
- ZFS
- OSS software stack currency
- size and scope of the repositories
- release parity
- 5 years base support
- LTS & non-LTS mix
- current OS and software every 6 months (non-LTS)
- new LTS OS / software every 2 years
- easy availability and trial
- same L&F like on other platforms
- Ubuntu KVM
- OpenStack integration
- MAAS KVM aka Pods
- no 31-bit legacy
- (HWE-) Kernel upgrade option within LTS
- Landscape client
- various repos/archives: Partner, UCA, PPAs
- PE support (d-i)
- ESM
- entry-level pricing (1-4 IFLs)
- ZFS
- Juju manual, local, MAAS provider
- MAAS
- Ubuntu Server OS, but also OpenStack and Kubernetes
- UA-I covers the Ubuntu Server OS, but also OpenStack and Kubernetes
- LTS & non-LTS mix
- LTS & non-LTS mix
- easy availability and trial
- same L&F like on other platforms
- IBM Java available from archives
Thank you - Questions?

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

- Canonical and IBM Webinar -- Ubuntu on POWER and LinuxONE
  https://youtu.be/b4VCU8M8-YA

- By the numbers: Ubuntu 16.04 LTS

- Creating the World’s Fastest OpenStack - Dustin Kirkland
  http://people.canonical.com/~kirkland/Using%20containers%20to%20create%20the%20World%27s%20fastest%20OpenStack%20%28public%29.pdf

- Container World 2016: Application and Machine Containers
  http://blog.dustinkirkland.com/2016/02/container-world-2016-application-and.html

- Ubuntu Server certified hardware

- Ubuntu 16.04 LTS for IBM LinuxONE and IBM z Systems is now available

- Ubuntu 16.04.x LTS Kernel Support Schedule
  https://wiki.ubuntu.com/Kernel/Support#A16.04.x_Ubuntu_Kernel_Support

- Ubuntu Linux 16.04 for IBM z Systems and LinuxONE - Installation Guide Wiki

- Fully- or Semi-Automated Installations (using preseed)
References and Links - ff - Canonical

- Infographic: Ubuntu Advantage explained
  https://insights.ubuntu.com/2016/10/25/infographic-ubuntu-advantage-explained/
- Ubuntu on Big Iron - Blog
  http://ubuntu-on-big-iron.blogspot.com/?view=sidebar
- Big Software has arrived!
  https://insights.ubuntu.com/2016/10/03/big-software-has-arrived/
- LXD Clusters: A Primer
  https://blog.ubuntu.com/2018/05/03/lxd-clusters-a-primer
- OpenStack on LXD
  http://docs.openstack.org/developer/charm-guide/openstack-on-lxd.html
- IBM & Canonical: A Virtualization and Cloud Computing (R-)Evolution
- Ubuntu Wiki – Features (incl. Security)
  https://wiki.ubuntu.com/Security/Features
- Ubuntu Wiki – Certification
- What’s new in Ubuntu 18.04 and OpenStack Queens
  https://www.brighttalk.com/webcast/6793/316587
References and Links - ff - IBM

- IBM LinuxONE and Ubuntu
- Ubuntu 16.04 Announcement (Mark Shuttleworth)
  http://www.ibm.com/systems/linuxone/open-source/ubuntu.html#s-1
- A Developer’s View: Why Ubuntu 16.04 on IBM LinuxONE (Marcel Mitran)
  https://www.youtube.com/watch?v=NDzofKWbd48
- Complimentary LinuxONE trial at IBM LinuxONE Community Cloud
  https://developer.ibm.com/linuxone/?source=web&ca=linuxone&ovcode=ov44223&tactic=C47300NW
- IBM and Canonical Ubuntu 16.04: Next-Generation Applications on LinuxONE (IDC Whitepaper)
- Validated Open-source Software for IBM Z and LinuxONE
  https://www.ibm.com/developerworks/community/forums/html/topic?id=5dee144a-7c64-4bfe-884f-751d6308dbdf
- IBM Z - Tested Operating System Platforms
  www.ibm.com/systems/z/os/linux/resources/testedplatforms.html
- IBM IT Infrastructure Blog / LinuxONE solutions Secure, scalable clouds and containers
References and Links - ff - Pervasive Encryption

- Pervasive Encryption - landing page: How to get started with pervasive encryption for IBM Z

- Linux encryption for data at-rest videos:
  - Pervasive Encryption for Data Volumes:
  - Setting up Data Volumes for Pervasive Encryption – In less than ten minutes:
    https://ibm.biz/BdzMs6 | https://youtu.be/t2Ph_h0LcsQ

- Pervasive Encryption for Data Volumes:
  http://public.dhe.ibm.com/software/dw/linux390/docu/l5n1dc03.pdf

- Getting Started with Linux on Z Encryption for Data At-Rest:

- Setting a master key on the cryptographic coprocessor:

- Security and Linux on IBM Z:
  http://www.redbooks.ibm.com/abstracts/redp5464.html
- Official Ubuntu Documentation
  https://help.ubuntu.com/

- Preseed installations - s390x specialties:
  Automated Ubuntu Server Installations on s390x with preseed

- Ubuntu for IBM Z and LinuxONE Wiki page (with FAQs, Bugs and hints)
  https://wiki.ubuntu.com/S390X
The Docu Chart(s) - Canonical - Ubuntu 16.04

- Ubuntu 16.04 LTS (Xenial Xerus) Server Guide

- Ubuntu 16.04 LTS (Xenial Xerus) Installation Guide
  - s390x (IBM System z):
  - recommended is to use the more current Wiki version of the Installation Guide for Ubuntu Server 16.04 for s390x:
    Ubuntu Linux 16.04 for IBM z Systems and LinuxONE - Installation Guide
  - and it’s special sub-page for preseed installations:
    Automated Ubuntu Server Installations on s390x with preseed
The Docu Chart(s) - Canonical - Ubuntu 18.04

- Ubuntu 18.04 LTS (Xenial Xerus) Server Guide

- Ubuntu 16.04 LTS (Xenial Xerus) Installation Guide
  - s390x (IBM System z):
The Docu Chart(s) - ff - IBM

- Documentation for Ubuntu Server 16.04 LTS
  https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_ubuntu.html
- Device Drivers, Features, and Commands on Ubuntu Server
  https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_udd.html
- Device Drivers, Features, and Commands on Ubuntu Server 16.04 LTS
- Device Drivers, Features, and Commands on Ubuntu Server as a KVM Guest
- Device Drivers, Features, and Commands on Ubuntu Server 16.04 LTS as a KVM Guest
- Troubleshooting Guide / Troubleshooting for Linux on z Systems (December 2016 edition, covering Ubuntu)
- The Virtualization Cookbook for IBM z Systems (IBM Redbook)
- IBM developerWorks - Hints for Canonical distributions (Ubuntu):
  https://www.ibm.com/developerworks/linux/linux390/distribution_hints.html#Canonical
The Docu Chart(s) - ff - Misc

- Hardware cryptography with Ubuntu Server on IBM Z and LinuxONE
  The paper is available via IBM Techdocs WP102721:
  Hardware cryptographic support for IBM Z and IBM LinuxONE with Ubuntu Server
  http://people.canonical.com/~fheimes/MG_HWCrypto_with_Ubuntu_on_z.pdf
- "linux-on-ibm-z/docs"
  collection of documents and tutorials to help you get open-source software up and running on Linux on z
  https://github.com/linux-on-ibm-z/docs/wiki
- IBM KVM to Ubuntu KVM (Migration document):
  https://wiki.ubuntu.com/IBMKVMtoUbuntuKVM
- Linux Channel Bonding Best Practices and Recommendations
  https://www.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.l0wlcb00/l0wlcb00_2018.html
  http://public.dhe.ibm.com/software/dw/linux390/docu/l0wlcb00.pdf
- Linux on z Systems and LinuxONE - Database as a Service
- Getting started with pervasive disk encryption
  http://public.dhe.ibm.com/software/dw/linux390/docu/l5n1dc00_a_quick_start.pdf
MAAS: Metal as a service*
MAAS automates the provisioning of hardware at cloud scale

1. **Improve data center agility**
   with blazing fast hardware provisioning

2. **Hardware discovery and inventory**
   for policy based management

3. **Orchestration of services**
   on bare metal hardware

* s390x: MAAS KVM only
Juju Service Orchestration

- **Simplify and automate** deployment and management of services
- **Service ubiquity** - provision on bare metal, private, and public clouds
- **Scale services** easily
- **Leverage** existing provision tools and language experience
- **Automation** ⏳ **Speed** ⏳ **Time to Market** ⏳ **Competitive Advantage**
Landscape

- The fastest way to **build a cloud**
- Advanced administration of OpenStack **infrastructure**
- Advanced administration of Cloud **guests and hosts**
- **Package & patch repository** management
- **Security, audit reports & compliance** control for Guests as well as Infrastructure

Landscape client available for s390x, Landscape Server as 'aaS' for s390x
Main Characteristics of snap(s)

- snaps (snap packages) is an easy to maintain and distribute everywhere packaging infrastructure
- snaps significantly reduce the complexity of packaging and providing updates
- snaps manages dependencies and compatibility with various libraries
- snap handling and processing is atomic, means software updates are "transactional"
- snaps are independent of the distribution or it’s release and backed by several Linux distributions
- snap is available with Ubuntu 16.04 LTS for all supported platforms, incl. s390x
- snaps are independent from the traditional package formats like, deb or rpm
- and can be used alongside traditional packages, deb format will continue to be supported
- the snap format, designed by Canonical, is handled by snapd, a free Github software project
- creating and maintaining packages can be complex and time consuming for ISVs and non experts
- it is distribution specific and it’s done rarely (e.g. major releases only), due to the required effort
- snaps enable vendors to distribute much faster, more often and broader, allowing more releases
- snaps are not only for desktops, but also for server, Cloud and IoT applications
- snaps enable (especially Cloud-) vendors to keep track with the speed of the web
- snaps are self-contained zip files that can be executed very fast in place
- each snap is confined using a range of Kernel isolation and security mechanisms (like containers)
- snaps integrate nicely into the operating systems by mounting squashfs ro
- snapcraft is a tool for development and the creation of snaps
- the snapcore team drives development of the project at snapcraft.io

https://insights.ubuntu.com/2016/06/14/universal-snap-packages-launch-on-multiple-linux-distros/
Fully- or Semi-Automated Installations

Preseed Overview

One of the Debian installers (d-i) advanced options is the automated install mode via preseed - a combination of special boot parameters and a preseed file.

The (d-i) installer on s390x runs in netboot mode and has two stages:
1) the initial stage (before networking has been started in the installer) usually runs at the console
2) and the second stage that is usually run via ssh.

Automating the installation can be done in multiple ways:
- Pass preseed keys or aliases via parmfile:
  e.g. debian-installer/locale=en_US domain=myhostname
- Append complete preseed.cfg to the initrd.ubuntu
- Preseed networking information via parmfile, and use url= parameter to fetch the rest of the config over the network

PARMFILE UBAAUTO

ro locale=en_US auto=true priority=critical apt-setup/proposed=true url=ftp://server/preseed/preseed.cfg
s390-netdevice/choose_networktype=qeth
s390-netdevice/qeth/layer2=true s390-netdevice/qeth/port=0
s390-netdevice/qeth/choose=0.0.0600-0.0.0601-0.0.0602 netcfg/get_ipaddress=10.245.236.12
netcfg/get_netmask=255.255.255.0 netcfg/get_gateway=10.245.236.1
netcfg/get_nameservers=10.245.236.1
netdevice/qeth/layer2=true netcfg/confirm_static=true
netcfg/use_autoconfig=1 netcfg/disable_dhcp=true
hostname=ZLIN12 domain=canonical.com network-console/start=note
network-console/password=your_password
network-console/password-again=your_password

Fully- or Semi-Automated Installations

preseed.cfg (example, partly)

```bash
# Fully- or Semi-Automated Installations

d-i debconf/priority string critical
# deactivate apt-get of the menu
#d-i auto-install/enable boolean true

d-i apt-setup/proposed boolean true
# use -proposed udebs

d-i pkgsel/update-policy select none
# Localization

d-i debian-installer/language string en

# activate qeth

# localize

d-i debian-installer/language string en

d-i s390-netdevice/choose_networktype string qeth

d-i s390-netdevice/qeth/choose string 0.0.0600-0.0.0601-0.0.0602

d-i s390-netdevice/qeth/layer2 boolean true

d-i s390-netdevice/qeth/port string 0

# use static network configuration

d-i netcfg/choose_interface select auto

d-i netcfg/disable_autoconfig boolean true

# activate DASD, dasdfmt if needed

d-i s390-dasd/dasd string 0.0.0200,0.0.0300,0.0.0400

d-i s390-dasd/auto-format boolean true

d-i s390-dasd/force-format boolean true

# DASD configuration

# auto-partition, all files in one partition

d-i partman-auto/method string regular

d-i partman-auto/choose_recipe select atomic

d-i partman/choose_partition select finish

d-i partman/confirm_nooverwrite boolean true

d-i partman/confirm boolean true

# user setup

d-i passwd/user-fullname string ubuntu

d-i passwd/username string ubuntu

d-i passwd/user-password-crypted

password $6$tYarsvlkZYGF$DiQv/Pr6JAXuIYUGQiESBMdnJApjquW0

# reboot at the end

d-i finish-install/reboot_in_progress note
```
