Linux in the Cloud, on Prem, or... on a Mainframe?

Elizabeth K. Joseph | @pleia2

Lyz@princessleia.com | lyz@ibm.com

15 January 2020

Linux.conf.au

Elizabeth K. Joseph (@pleia2)

Developer Advocate, IBM Z

(I talk to techies, I don't know how to sell you a mainframe 🤷)

Debian / Ubuntu

OpenStack

Apache Mesos

Linux on Z

But mostly I'm a systems administrator.

Distribute!

Cloud!

OpenStack!

CI/CD!

Startups!

Disrupt!

Kubernetes!

So, in the cloud or on premises?

x86 as far as the eye can see.

But there are other things out there! ARM, Power, RISC-V, IBM Z (s390x... mainframe!)

What is a mainframe?

IBM System 360 (s/360), 1964

IBM z15, 2019

What is a mainframe?

Depends on who you ask.

Traditionally runs z/OS, but increasingly Linux too.

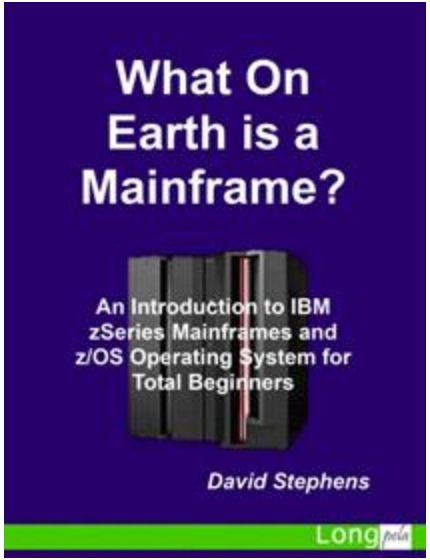
Data, data, data.

Batch processing!

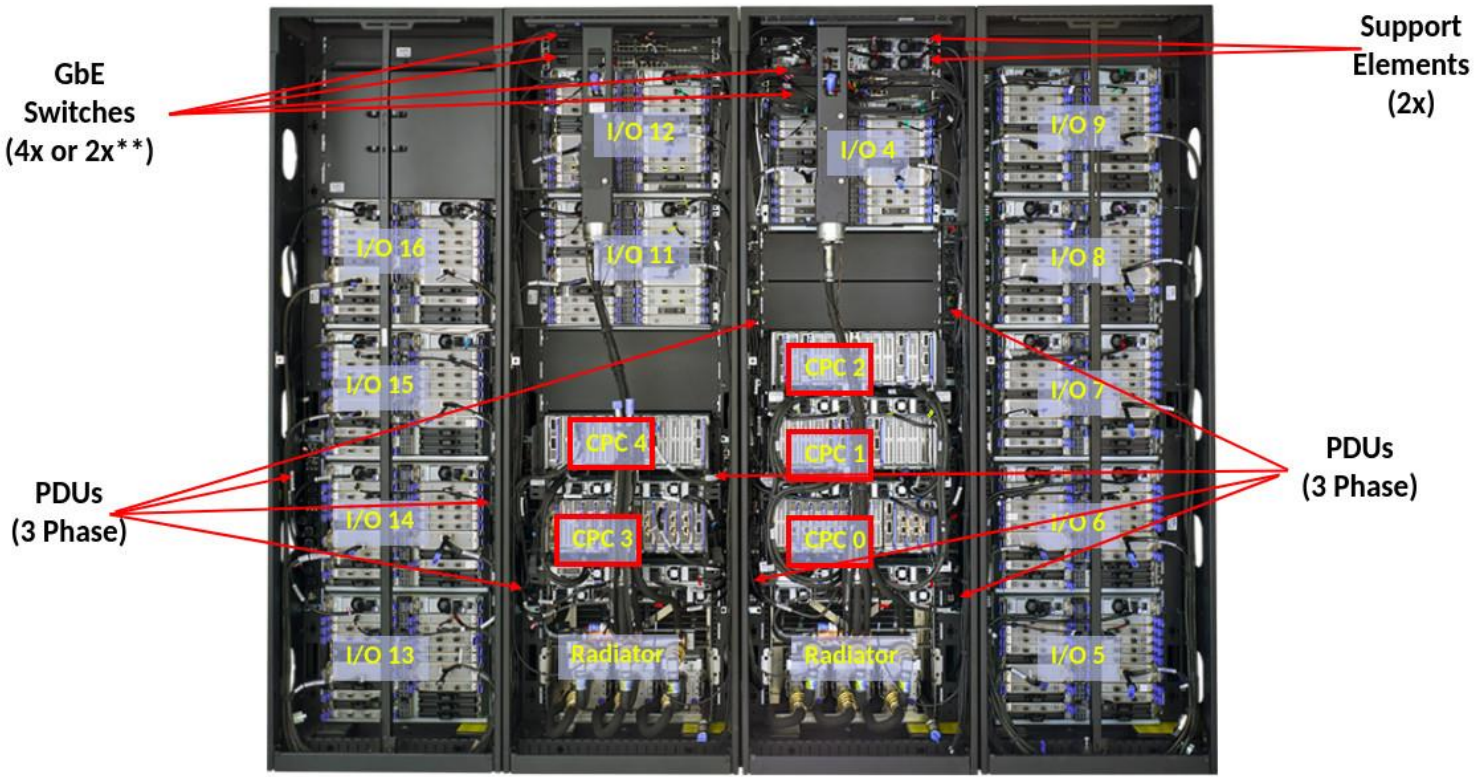
Enterprise-grade hardware and external storage.

Virtualization? Solved!

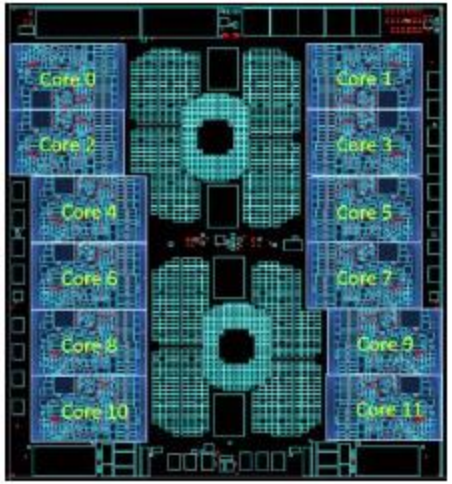
Networking? Solved!



What is a mainframe?



What is a mainframe?



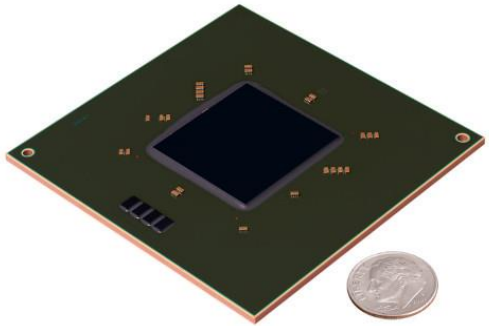
Not x86.

(IBM Z | zArchitecture | s390x)

190 5.2 ghz processor units (PUs), with 12 cores per chip

But also...

- 40TB of RAM
- 60 PCIe control units across 12 PCIe I/O drawers
- 22 dedicated I/O offload processors (SAPs) pre-allocated per system
- https://developer.ibm.com/blogs/systems-inside-the-new-ibm-z15/



Storage - DS8900F

The highest end model, the IBM DS8950F Model 996 has nearly 5.9 PB (5,898 TB) maximum physical capacity

But also...

Storwize® V5100/F

Storwize V7000

FlashSystem® 9100

FlashSystem 900

DS8882F

So, what runs on it?

z/OS

z/OS, a widely used mainframe operating system, is designed to offer a stable, secure, and continuously available environment for applications running on the mainframe.

z/VM

As a control program, z/Virtual Machine (z/VM) is a hypervisor because it runs other operating systems in the virtual machines it creates.

z/VSE

z/Virtual Storage Extended (z/VSE) is popular with users of smaller mainframe computers. Some of these customers eventually migrate to z/OS when they grow beyond the capabilities of z/VSE.

z/TPF

The z/Transaction Processing Facility (z/TPF) operating system is a special-purpose system that is used by companies with very high transaction volume, such as credit card companies and airline reservation systems.

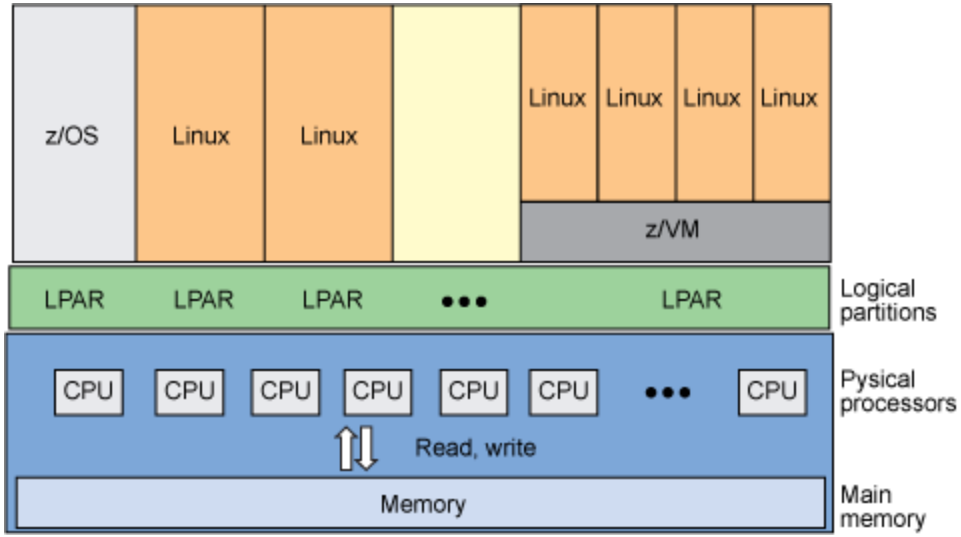
Linux for System z

Several (non-IBM) Linux distributions can be used on a mainframe.
Source:
https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmainframe/zconc_opsysintro.htm

Modern mainframes run Linux!

...and they have for 20+ years

How it works with Linux



There is always some kind of virtualization being used for Linux on Z.

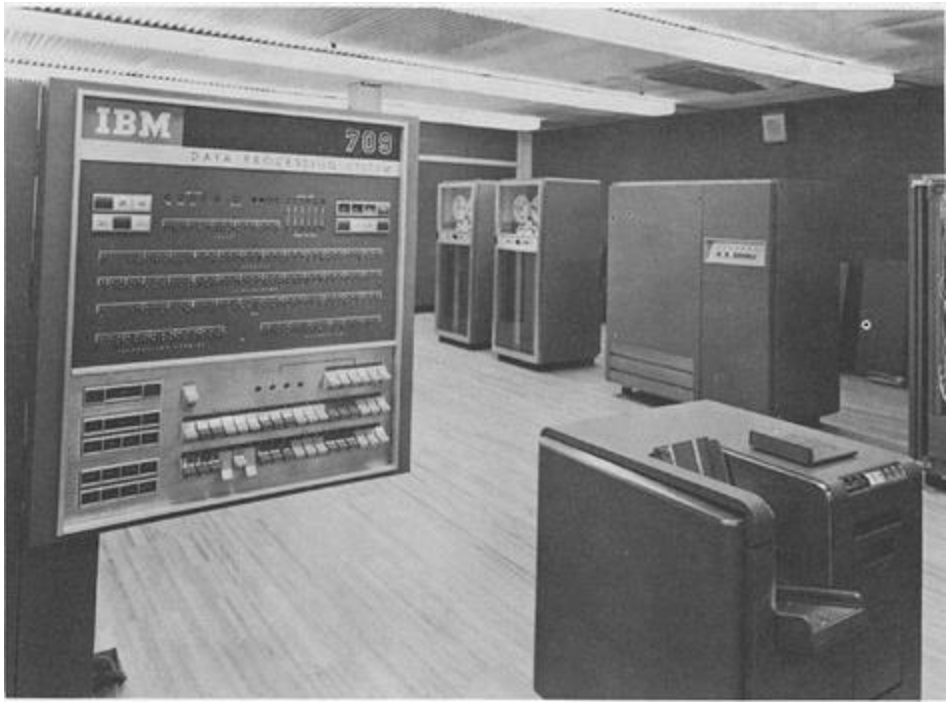
Using z/VM (or KVM!), one or more Linux installs can be put on a single Logical Partition (LPAR).

Using Processor Resource and System Manager (PR/SM) a single Linux instance can be installed on a single LPAR.

Image source: https://www.ibm.com/developerworks/library/l-systemz/

Once upon a time mainframes lacked time-sharing

Papers discussing time-sharing were published as early as 1959, but Compatible Time-Sharing System (CTSS) was first demoed by MIT on an IBM 709 in 1961.



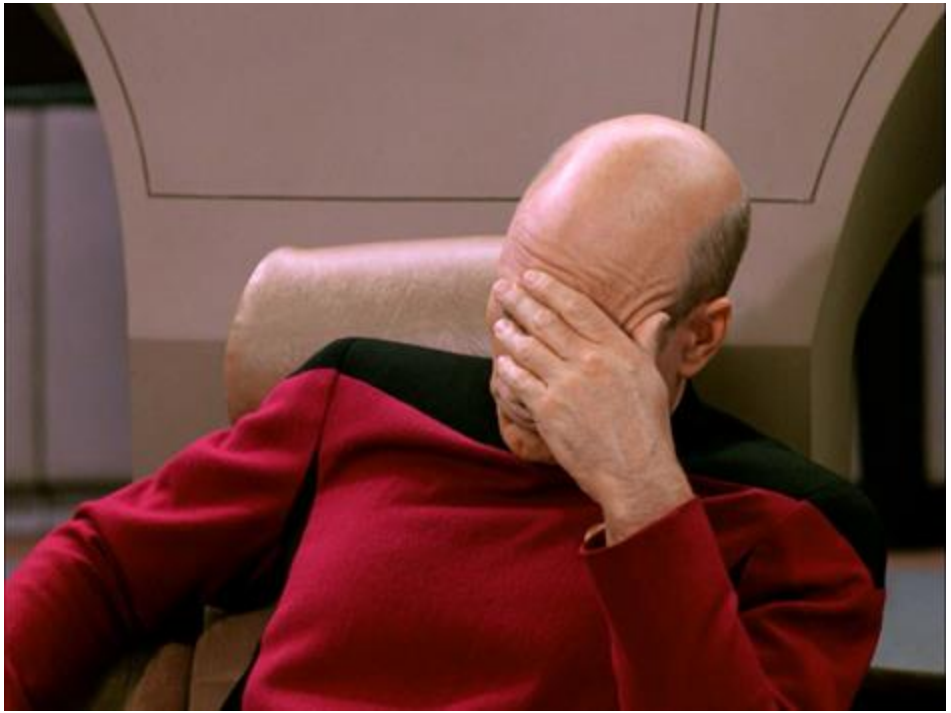
Several iterations later… VM/370, in 1972

Want to know about all those iterations? Melinda Varian has published a fascinating history, available in several formats, on her website: http://www.leeandmelindavarian.com/Melinda/

IBM: **“I don’t think anyone needs VMs”**

(paraphrased)

The Doubtful Decade.



But it got better

The Doubtful Decade ended and VM community thrived, along with the technology and support from IBM.

In 1994 experimental TCP/IP support was added to VM, adding a key component to supporting Linux 5 years later.

Linux Origins: Bigfoot

Developed by Linas Vepstas in 1998-1999 as a community effort.

“the **Bigfoot (i370)** port was started first, but is currently stagnant for essentially political, social, and market reasons.”

Source: Linas Vepsta’s site on Linux on s390
https://linas.org/linux/i370.html

Why did the community want it?

“Why? Good question. One we've asked ourselves many times. Why do you do the things you do? If you think about it, you can probably find a hundred rationalizations for what your gut makes you to do. Here's some of ours:

- Stunt
- To Learn
- Because Its There
- Because Its Knarly, Duude!
- I/O
- Address Spaces and Access Lists
- VM
- The Business Model”

Source: https://linas.org/linux/i370-why.html

Linux Origins: Linux for S/390

Linux for S/390 began when “IBM published a collection of patches and additions to the Linux 2.2.13 kernel on December 18, 1999, to start today's mainline Linux on Z. Formal product announcements quickly followed in 2000”

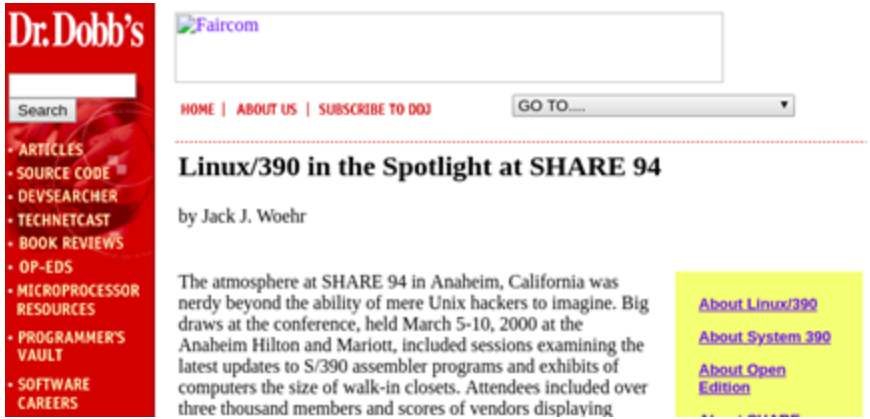
Marist File System was the first Linux distro put together out of Marist College in Poughkeepsie, NY. Think Blue Linux by Millenux in Germany was an early distro with Red Hat packages and the IBM kernel for mainframes. Other commercial editions quickly followed.

Source: https://en.wikipedia.org/wiki/Linux_on_z_Systems

This is the current, actively developed iteration that all the major platforms are part of today.

Why did IBM want it?

IBM “Heist” commercial, 2001 https://www.youtube.com/watch?v=uxg17JlyFas



ComputerWorld (Denmark): Linux on IBM S/390 mainframe

Oct 12, 1999, 01:52 UTC (19 Talkbacks) (Other stories by J.O.S. Svendsen)

[Linux Today reader Hans Schou writes:]

"Friday 8 october 1999 there was a story in the Danish Computerworld about IBM had ported Linux to the S/390 mainframe.

For some people this would not be amazing, as there was a posting to the Linux Kernel list back in march 1999, where a guy asked about DMA buffers and address space. The posting came from 3dlabs.com and the rumor began that IBM was porting Linux to mainframe.

I called IBM today and they confirmed that the development was going on, but they did

20 Years of Linux

Networking

Between Linux LPARs, HiperSocket is used for communication between VMs rather than TCP/IP for speed, responsiveness and reliability.

Storage

Linux can connect and interface with to the storage servers, like the DS8900F.

Portions of Linux rewritten to take advantage of hardware I/O capabilities reducing load from the Central Processor (CP).

Processors

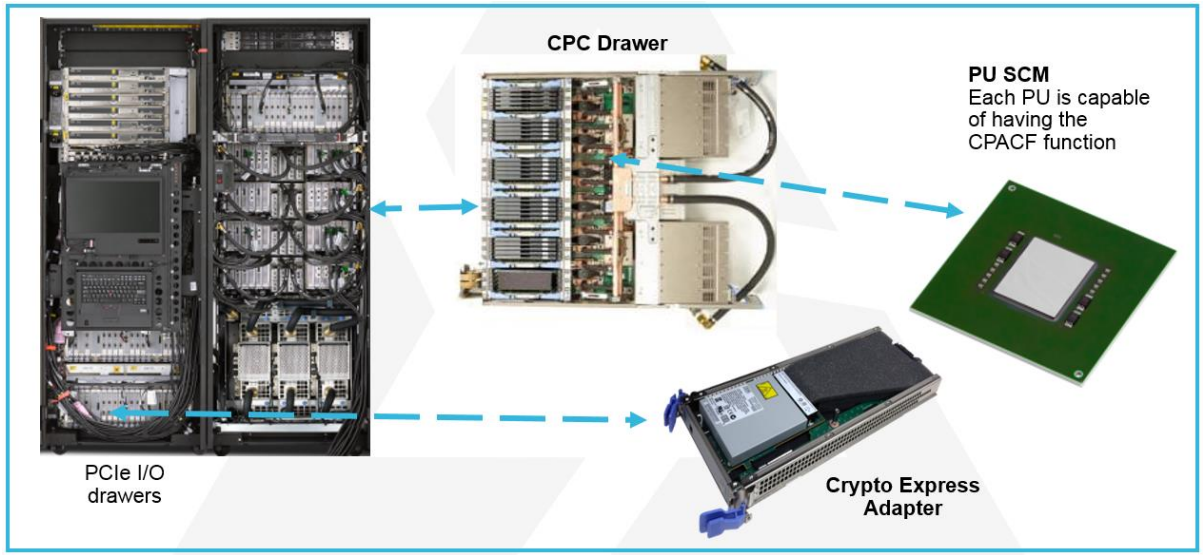
Linux can run on the traditional mainframe Central Processor (CP), but there's also an Integrated Facility for Linux (IFL) processor with some instructions disabled that are used only by z/OS.

Open Source

There are few barriers to compiling for s390x (though it is big-endian), so new open source software is being compiled for the platform every day.

Validated open source software list: https://www.ibm.com/community/z/open-source-software/

Encryption

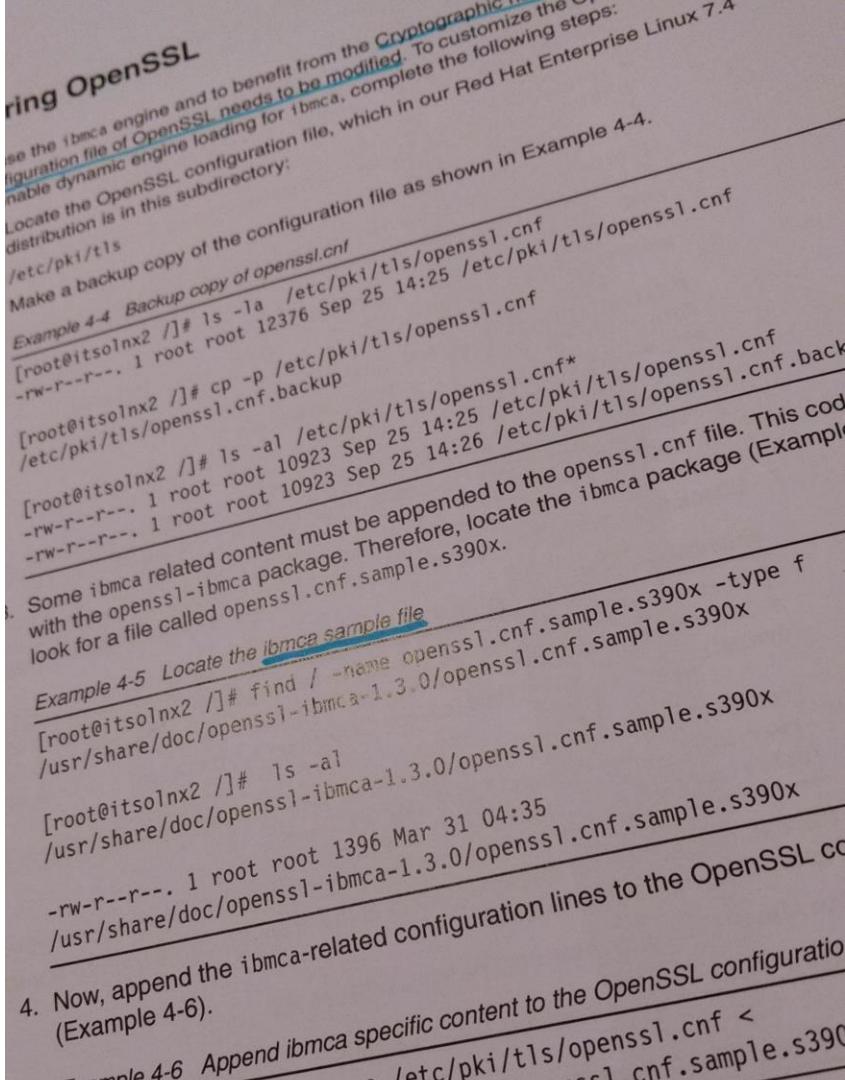

CPC Drawer
PU SCM
Each PU is capable of having the CPACF function
PCIe I/O drawers
Crypto Express Adapter

How it's used on Linux

Security for Linux on System Z
http://www.redbooks.ibm.com/abstracts/sg247728.html

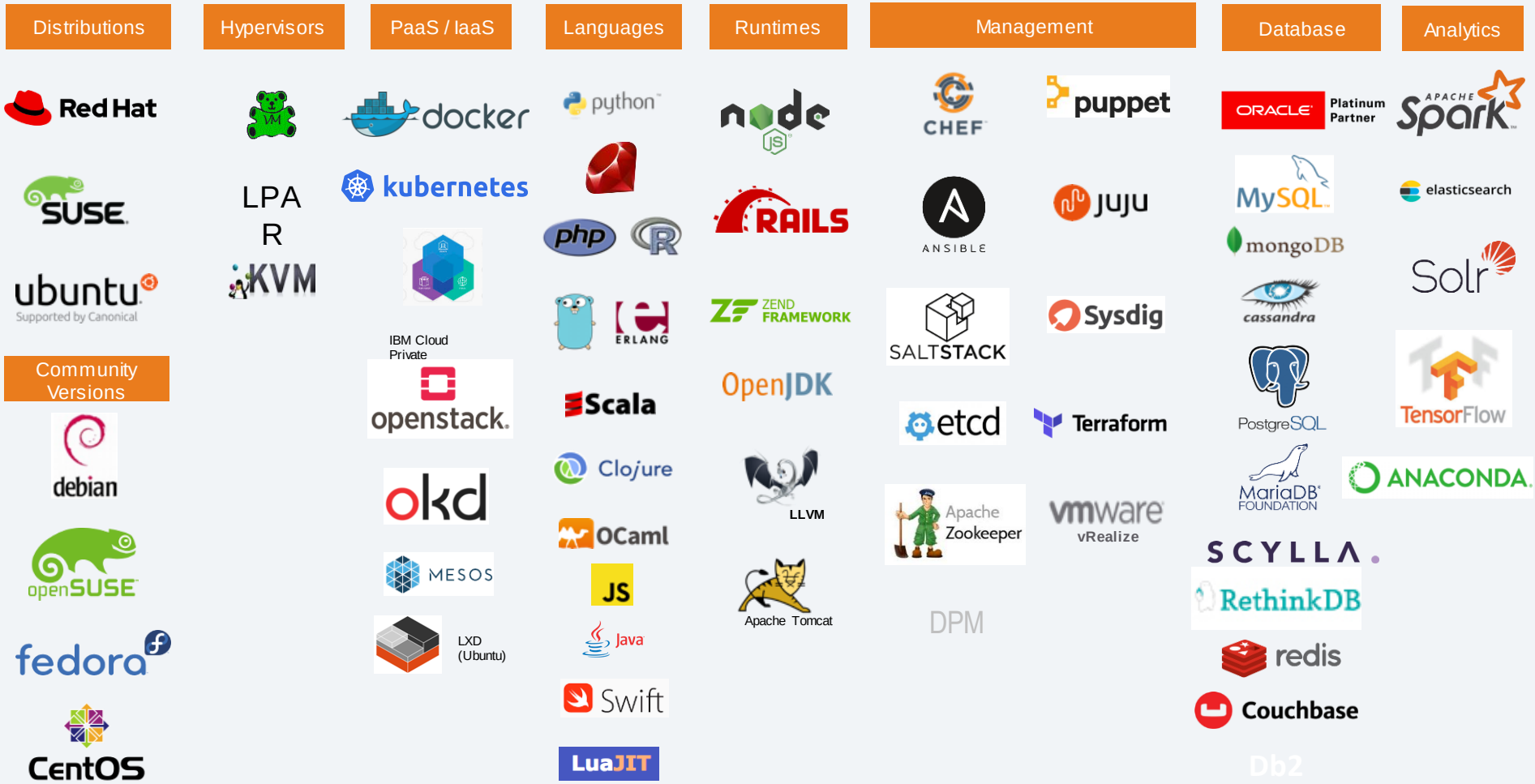
- dm-crypt
- OpenSSL and libcrypto (including for ssh, scp, sftp, Apache mod_ssl...)
- IPSec
- Built-in encryption in Java and Go

And the open source libica crypto library for s390x https://github.com/opencryptoki/libica



LinuxONE

2019, LinuxONE III (z15 with IFLs)


Distributions
Hypervisors
PaaS / IaaS
Languages
Runtimes
Management
Database
Analytics
Red Hat
SUSE
ubuntu
Supported by Canonical
Community Versions
debian
openSUSE
fedora
CentOS
LPAR
KVM
docker
kubernetes
IBM Cloud Private
openstack.
okd
MESOS
LXD (Ubuntu)
python
php
ERLANG
Scala
Clojure
OCaml
JS
Java
Swift
LuaJIT
node
RAILS
ZEND FRAMEWORK
OpenJDK
LLVM
Apache Tomcat
CHEF
puppet
ANSIBLE
juju
SALTSTACK
Sysdig
etcd
Terraform
Apache Zookeeper
vmware vRealize
DPM
ORACLE Platinum Partner
APACHE Spark
MySQL
elasticsearch
mongoDB
Solr
cassandra
PostgreSQL
TensorFlow
MariaDB FOUNDATION
ANACONDA
SCYLLA
RethinkDB
redis
Couchbase
Db2
THE LINUX FOUNDATION
OPEN MAINFRAME PROJECT

Your software?

Build it for the mainframe!

Unleash the power to innovate with IBM **LinuxONE**™ Community Cloud

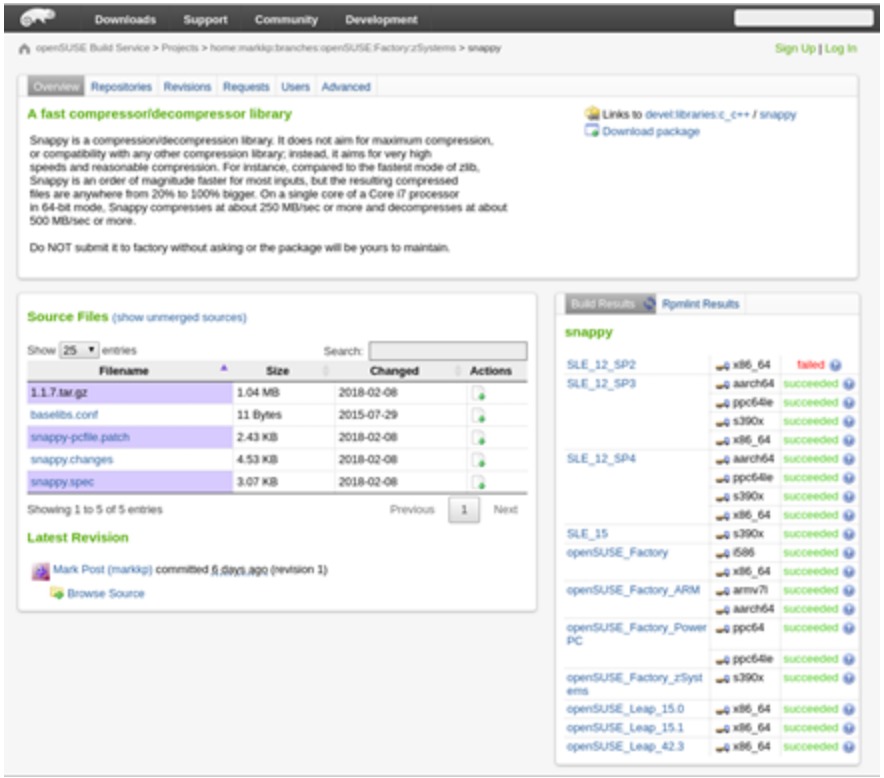
The IBM LinuxONE Community Cloud provides a no charge, self-provisioned SUSE or Red Hat virtual machine on an IBM LinuxONE Enterprise Server (s390x architecture) to develop, test and run your apps.

https://developer.ibm.com/linuxone

Self-Service s390x: Ubuntu PPAs

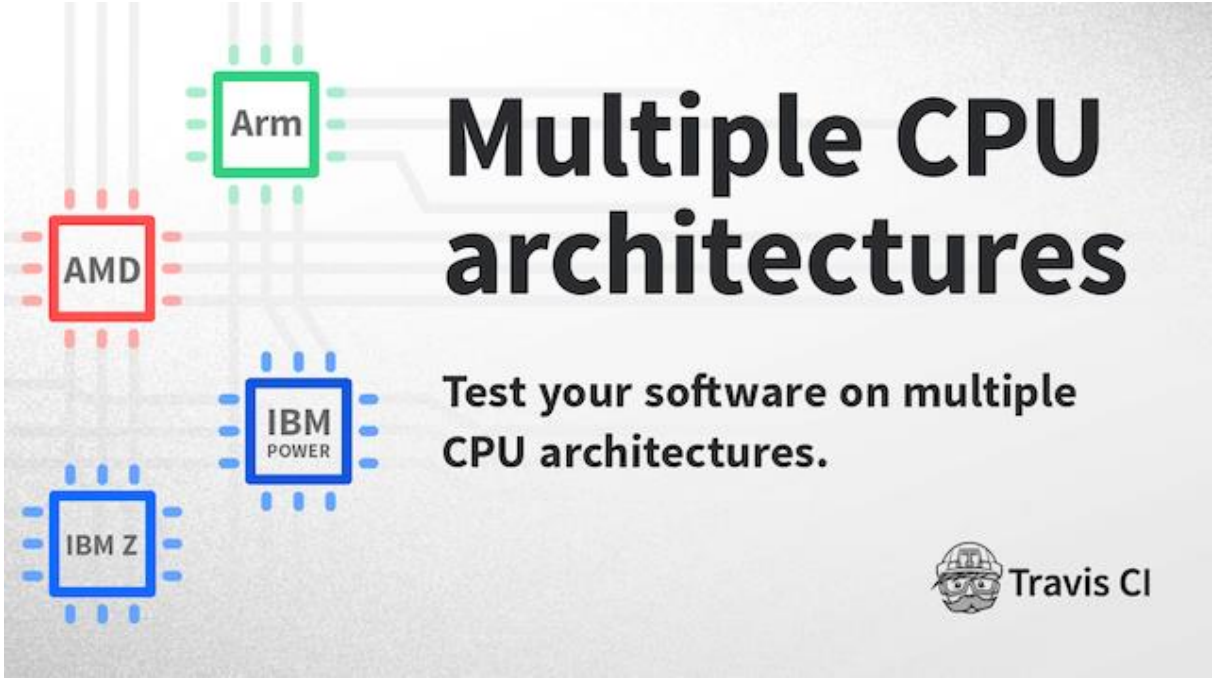
See: https://princessleia.com/journal/2019/06/building-a-ppa-for-s390x/

Self-Service s390x: openSUSE Build Service



Source: https://build.opensuse.org/

Self-Service s390x: Travis CI



See: https://blog.travis-ci.com/2019-11-12-multi-cpu-architecture-ibm-power-ibm-z

Cryptography in the Cloud, powered by mainframes

IBM Cloud Hyper Protect Services: Crypto, DBaaS, Virtual Servers, and Containers (soon)

https://www.ibm.com/cloud/hyper-protect-services

IBM Blockchain Platform

https://www.ibm.com/cloud/blockchain-platform

HYPERLEDGER FABRIC

Kubernetes

SUSE
https://developer.ibm.com/storage/2019/03/01/kubernetes-1-12-on-suse-linux-using-kubeadm/

Red Hat
https://www.ibm.com/blogs/systems/get-ready-for-red-hat-openshift-on-ibm-z-and-linuxone/

Ubuntu
https://ubuntu-on-big-iron.blogspot.com/2019/08/deploy-cdk-on-ubuntu-s390x.html

Sine Nomine Associates with OpenShift Origin
https://www.sinenomine.net/products/linux/OpenShift

ICU IT Services
"integrating their traditional zOS environments with new (private) cloud environments."

Orchestration across your fleet with the huge Kubernetes tooling ecosystem.

Run the same workloads, with the same tools, on premises and in the cloud.

Integration with traditional z/OS environments, such as running containerized workloads close to their large data environments (DB2 on z/OS or Oracle on Linux on z) to reduce latency.

Working with z/OS

Open Mainframe Project projects:
https://www.openmainframeproject.org/projects

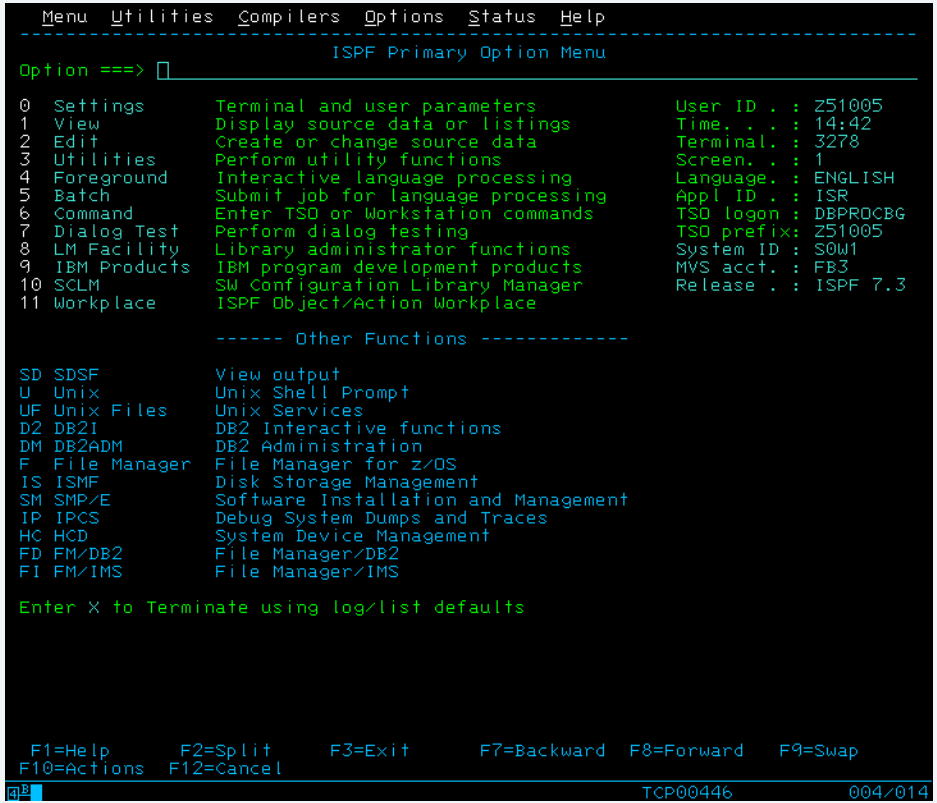
Zowe website: https://www.zowe.org/

Zowe Overview (November 2018):
https://www.youtube.com/watch?v=NX20ZMRoTtk

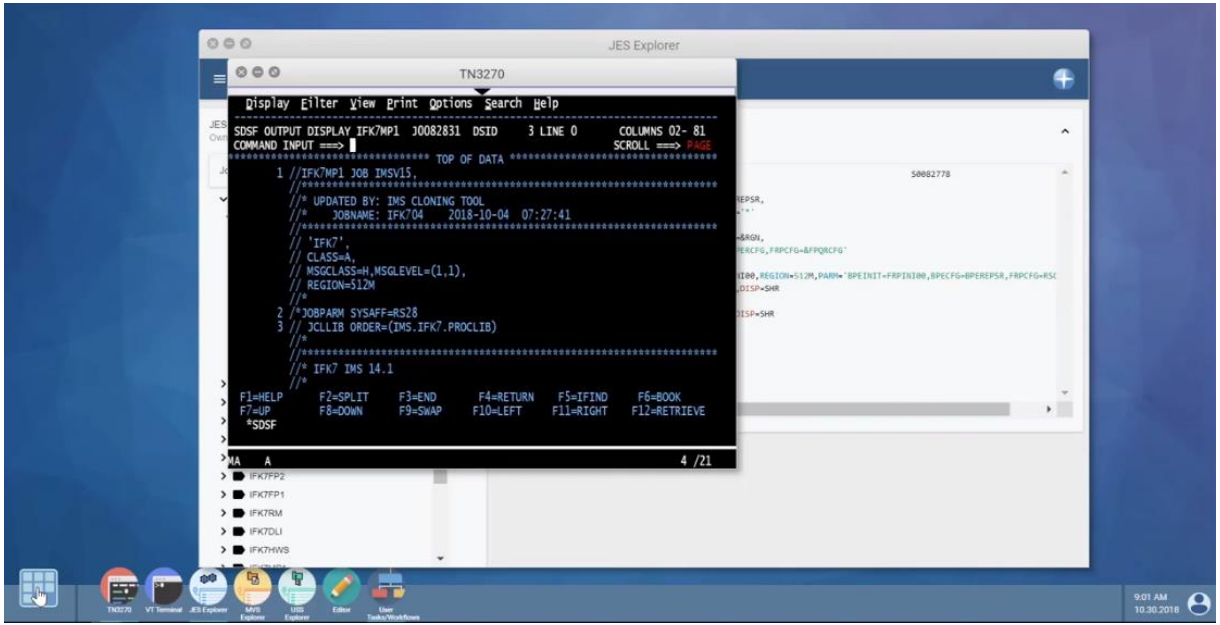
Zowe Webinar (22 February 2019):
https://www.youtube.com/watch?v=XixEltbRmds

Traditional interaction: ISPF

(Interactive System Productivity Facility)



Modern interaction: Zowe Web Desktop



- Desktop-like environment, in your web browser
- Includes a 3270 terminal
- Provides GUI-based explorers for files and datasets with ways to manipulate data

Modern interaction: Zowe CLI

Modern command line tool

- Execute zowe commands from standard MacOS/Linux terminal
- Includes core commands for interacting with mainframe-specific needs (TSO, batch processing)
- Extendable for your specific needs, or those of vendors who wish to offer support for their tooling

Can build IDE extensions, a Visual Studio Code extension already exists.

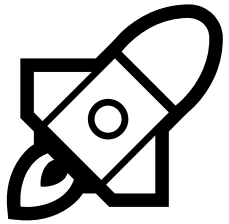
Automation and scripting, including CI/CD pipelines!

Example usage, Using the Zowe CLI to edit a data set:

https://docs.zowe.org/stable/user-guide/zowe-getting-started-tutorial.html#using-the-zowe-cli-to-edit-a-data-set

Modern interaction: Zowe API

- API gateway that can abstract out the various number of services running on the mainframe that you may wish to interact with: job services, dataset services, and more
- Open source and vendor products are now leveraging the API for new products that interact with the mainframe

The Future! 

Energy Savings

Paul Newton: The world can’t take all the x86 machines we’ll need at the rate server farms are growing.

Me: Cost or, like, the planet?

PN: Yes.

We will encrypt more stuff

Need for fewer impactful data breaches (stolen data is useless if it’s encrypted!)

More laws and regulations around access and use of customer data.

Further Architecture Diversification

We’re now seeing an increase in hardware architectures, not a decrease!

Every year we have more compelling reasons to not solely depend upon one architecture

The title of this talk

is a false dichotomy.

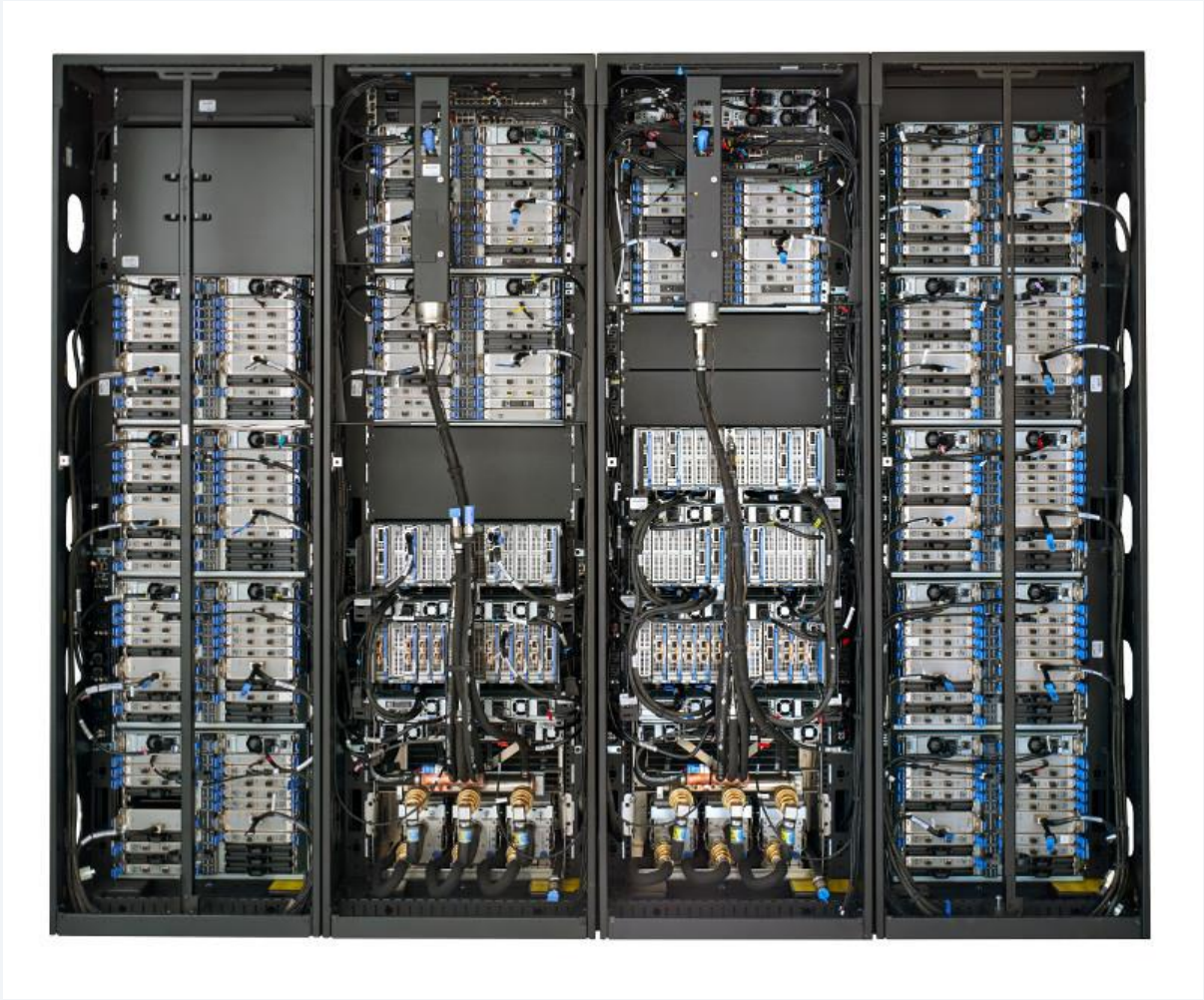
You don't need to

choose.

Questions?

Elizabeth K. Joseph

Developer Advocate, IBM Z

—

Twitter: @pleia2
Email: lyz@ibm.com