Positioning Linux and Linux Application Cloud Deployment Options for IBM Z and IBM LinuxOne Platforms

IBM

This document can be found on the web at www.ibm.com/support/techdocs

Version Date: November 2017

Mike Bonett

IBM - Washington Systems Center (bonett@us.ibm.com)

Special Notices

This document reflects the IBM Washington Systems Center's understanding on IBM Linux z Cloud Options. It was produced and reviewed by the members of the IBM Washington Systems Center organization. This document is presented "As-Is" and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the IBM Washington Systems Center. These opinions are based on hands on experiences with the listed technologies. If you have questions about the contents of this document, please direct them to Mike Bonett (bonett@us.ibm.com).

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries: Bluemix, IBM Z, IBM LinuxONE, System z, WebSphere, z/OS, z/VM, z Systems. A full list of U.S. trademarks owned by IBM may be found at https://www.ibm.com/legal/us/en/copytrade.shtml.

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.
LINUX and Linux are registered trademarks of Linus Torvalds in the United States, other countries, or both. .
KVM and the KVM logo are trademarks or registered trademarks of KVM.
Java and all Java-based trademarks and logos are registered trademarks of Oracle and/or its affiliates.
Red Hat and the Shadowman logo are registered trademarks of Red Hat, Inc. in the U.S. and other countries.
VMWare and the VMWare logo are trademarks or registered trademarks of VMware, Inc. in the United States and other countries.
Ubuntu and the Ubuntu logo are trademarks or registered trademarks of Canonical, Inc. in the United States and other countries.
Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and other countries.
SUSE, SLES, and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries
Other company, product and service names may be trademarks or service marks of others.

Contents

Introduction

The IBM Z and IBM LinuxOne platforms provide the virtualization and management capabilities to support a Linux cloud environment. A "Cloud Environment" adheres to the industry standard characterizes of a Cloud Computing Delivery Service:

- **On-Demand Self Service** – allows the selection of services when needed, typically through some type of services catalog
- **Broad Network Access** - available to users via network connectivity, both thin or thick clients
- **Resource Pooling** - resources are shared and serve multiple consumers
- **Rapid Elasticity** - capabilities provisioned, in some cases automatically, and can be scaled as needed or requested
- **Measured Service** –ability to measure usage and have users pay for what they use
- **Multi-tenancy** - ability to isolate users and groups across the shared infrastructure resources

For Linux, IBM Z and IBM LinuxONE provide the technology platform to enable a cloud computing environment with these characteristics.

Once the supporting infrastructure is in place, the level(s) of cloud services to be provided can be determined. The most common ones, from a Linux perspective, are:

- **Infrastructure as a Service (IAAS)** – provide Linux compute platforms that users can deploy.
- **Platform as a Service (PAAS)** – Provide Linux compute platforms with software (usually called middleware) which users can deploy to develop and build applications.
- **Software as a Service (SAAS)** – Provide Linux based applications that users can deploy and use, without concerning themselves with the underlying computer and middleware platform.

While there are other xAAS options, they tend to be specialized derivatives of the above three. For example, Container as a Service (CAAS) can be viewed as a specialized example of either PAAS or SAAS, depending on the contents of the software container.

Cloud management entails, among other things, managing the user access to the services and invoking functions to carry out user requests for the services. There are two considerations to considering when applying this specific management aspect for cloud services to IBM Z and LinuxOne:

- **Manage-from**: Cloud management is done from the IBM Z or LinuxOne platform. The functions to drive automated deployment of the services reside on the platform.
- **Manage-to**: Cloud management is performed outside of the IBM Z and LinuxOne platforms. The functions to drive end user access and automated deployment of the services reside interface into the IBM Z and LinuxOne platforms to invoke functions to carry out the activity.

This distinction is made because some environments may have a preference for managing the cloud environment on IBM Z or LinuxOne from IBM Z or LinuxOne.

From a cloud perspective, there are several different technology options for using the IBM Z and LinuxOne platforms to support cloud computing services. The desired types of cloud services to be provided, the desire for where the services will be managed from, and the requirement for how the IBM Z and LinuxOne will be managed, will influence the technology selection. These are not the only items to consider – price, skills, time

to value, etc. are also important - but this paper will focus on them as they are a good starting point, against which the other considerations can be applied.

Linux Distribution Support Considerations

Currently there are three vendors who provided supported Linux distributions for the IBM Z and LinuxOne platform:

- Red Hat
- SUSE
- Canonical (Ubuntu)

Not all of these distributions are supported in every option listed below. While this white paper will highlight some of the key distribution support considerations for each option, **this is a very dynamic area subject to change. It is important to go to the detailed links provided for each option to get the latest information on distribution support for the option being investigated.**

Technology Options Summary

The table below is a summary of the available options, the primary usage of the options in the cloud context, the primary cloud service level(s) the option provides, and the manage-from or manage-to support that is available for the option:

Option	Purpose (cloud perspective)	Primary Service Levels Provided	Manage-from IBM Z and IBM LinuxOne?	Manage-to IBM Z and IBM LinuxOne?
IBM Dynamic Partition Manager (DPM)	Programmatically deploy z/VM LPARs running Linux or KVM for z	IAAS	Depends on OpenStack platform location	via OpenStack APIs
z/VM Cloud Manager Appliance (stabilized)	Self-service deployment of Linux guests under z/VM via OpenStack	IAAS, custom PAAS, custom SAAS	Yes (for z/VM LPARs)	via z/VM CMA
z/VM Cloud Enablement APIs (future)	Allow z/VM to be integrated into OpenStack and non-OpenStack environments for cloud deployment	IAAS	Depends on management software support for z platform execution	via APIs
KVM on IBM Z	Self-service deployment of Linux guests under a KVM LPAR	IAAS, custom PAAS, custom SAAS	Yes	Via OpenStack

Option	Purpose (cloud perspective)	Primary Service Levels Provided	Manage-from IBM Z and IBM LinuxOne?	Manage-to IBM Z and IBM LinuxOne?
IBM Wave	Administrative cloning of Linux guests under z/VM	IAAS	Yes	Via IBM Wave UI
Kubernetes	Open source technology for deploying containerized (aka "cloud native") applications	PAAS, SAAS	Yes	Yes
IBM Cloud Private	Self-service deployment of applications and microservices in Docker containers in a Kubernetes environment	PAAS, SAAS (IAAS via IBM Cloud Automation Manager)	No	Yes (Linux on z System guests are worker nodes to support containers)
IBM LinuxOne Community Cloud	Public cloud that provides free of charge Linux on z System environments for a limited time, to verify and test new apps and try new technologies	IAAS	Yes	via z/VM CMA

All of these options, with the exception of the LinuxOne Community Cloud, are meant for on premise usage. The Linux Community Cloud is included to position it with the other options.

The balance of this paper will provide a brief description of each option and expand on the cloud services and management contexts described above, as related to IBM Z and LinuxONE, for each option.

IBM Dynamic Partition Manager (DPM)
The IBM Dynamic Partition Manager (DPM) provides a user interface that is designed to support a quick deployment of an IBM Z (z13 and z14) or LinuxOne LPAR with an operating system (e.g. Linux, KVM, IBM Secure Services Container). It is particularly designed for setting up Linux servers via the IBM Hardware Management Console (HMC), but with a much simpler and guided interface that does not require in-depth knowledge of HMC LPAR configuration details.

DPM automatically discovers and displays the system resources that are available for deployment. It can also be used after the Linux server is running, to monitor system resources, define alter conditions, and adjust LPAR resources.

More information on DPM can be found at these links:

- IBM Dynamic Partition Manager Guide
  (http://www.ibm.com/support/docview.wss?uid=isg2c29b5c70ef6ddd178525803d00671411)
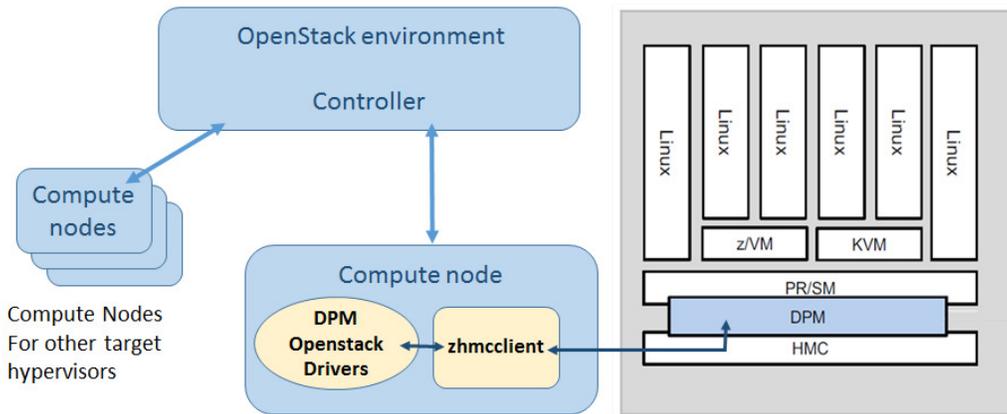
Cloud Services Context

The HMC Web Services application programming interface (HMC API) supports calls to invoke DPM functions. These can be used to programmatically deploy and manage DPM Linux LPARs. An open source implementation of an HMC API client with DPM support, called zhmcclient, is available at https://github.com/zhmcclient/python-zhmcclient, with documentation at https://media.readthedocs.org/pdf/python-zhmcclient/latest/python-zhmcclient.pdf . These APIs can be used to build a self-service interface for deploying and managing DPM partitions.

OpenStack drivers that support DPM for deploying instance are supported in the OpenStack Ocata and later releases. The driver code and documentation is available at the following links:

- https://github.com/openstack/nova-dpm
- https://pypi.python.org/pypi/nova-dpm
- http://nova-dpm.readthedocs.io/en/latest/

The following picture is one example of DPM integrated with an OpenStack architecture:

The drivers would be installed on an OpenStack compute node platform, and carry out the DPM functions directed by OpenStack via the zhmcclient API, which must also be installed on the compute node platform.

The cloud services most likely to be used for DPM are IAAS. The DPM focus is on getting a Linux Server up and running quickly, but any further software configuration on the server will require additional customization.

Management Context

Since DPM resources in a cloud context are managed via the HMC API, it can be managed from any platform that supports using the API, including IBM Z and Linux One. For OpenStack management, any platform that supports an OpenStack compute node at the Ocata or later release can manage a DPM environment.

z/VM Cloud Manager Appliance (CMA)

The z/VM Cloud Manager Appliance (CMA) integrates z/VM into an OpenStack environment to deploy Linux guests under z/VM. It provides a subset of the xTreme Cloud Administration Toolkit (xCAT), OpenStack services, OpenStack drivers, and an OpenStack API for use by OpenStack and non-OpenStack environments to integrate with.

The z/VM CMA requires DIRMAINT (or an equivalent support directory manager) and the z/VM Systems Management Application Programming Interface (SMAPI) to be enabled on z/VM. The CMA is fully supported in z/VM 6.3 (until z/VM 6.3 goes out of service the end of 2017) and z/VM 6.4.

More details on the CMA can be found at the following links:

- z/VM OpenStack Enablement : http://www.vm.ibm.com/sysman/openstk.html
- z/VM Cloud Manager Appliance Implementation Options:
  https://www..ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106386

IBM also provides OpenStack z/VM drivers, which can be used in other OpenStack environments to target the CMA for deploying Linux guests under z/VM. The drivers and documentation are available at the following links:

- Nova (Compute): https://github.com/openstack/nova-zvm-virt-driver
- Neutron (Network): https://github.com/openstack/networking-zvm
- Ceilometer (Measurement/Billing/Events): https://github.com/openstack/ceilometer-zvm

IMPORTANT NOTE: On October 10 2017, IBM announced that the z/VM CMA has been stabilized (https://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS217-516&appname=USN). In sum, this means:

- The code is stabilized at the OpenStack Newton level. No further upgrades or enhancements beyond this level will be provided.
- It will continue to be a supported component of z/VM; IBM will continue to provide support for client problems and provide security fixes.
- It no longer ships as a DVD image with z/VM 6.4, but can be requested from the IBM Support Center by anyone with a z/VM 6.4 license.

The z/VM cloud enablement APIs   (described in the next section) will become the primary option for integrating z/VM into OpenStack or other virtualization management platforms. However, depending on requirements, the CMA can still be a viable option for a Linux on z cloud environment.

Cloud Services Context

There are several different options for configuring z/VM CMA to use OpenStack services to deploy and manage Linux guests:

- As a standalone z/VM OpenStack environment. This is the recommended initial implementation. The CMA runs in "OpenStack controller mode" on a LPAR with the OpenStack z/VM drivers and can deploy

guests on that LPAR as well as other connected LPARs which have their CMA configured as OpenStack compute nodes. In addition, non-OpenStack applications such as VMware vRealize Automation can use OpenStack APIs to deploy Linux guests.

- Integrated into another OpenStack environment that contains the OpenStack Stack controller. The CMA can either:
    1. Run as OpenStack Compute nodes on each LPAR where z/VM guests are to be deployed.
    2. Be a target of the other OpenStack environment, which would use the z/VM drivers in its compute nodes to deploy and manage z/VM Linux guests. In this case the CMA would only run xCAT functions. SUSE OpenStack Cloud version 6 is an example of an OpenStack environment that supports this configuration.

For all of these configurations the IAAS cloud service can be provided, where the infrastructure is virtualized guests, storage, and networking under z/VM. These environments also support custom PAAS and SAAS services via the Heat component for OpenStack, or the VMware OpenStack endpoint for VMware vRealize Automation.
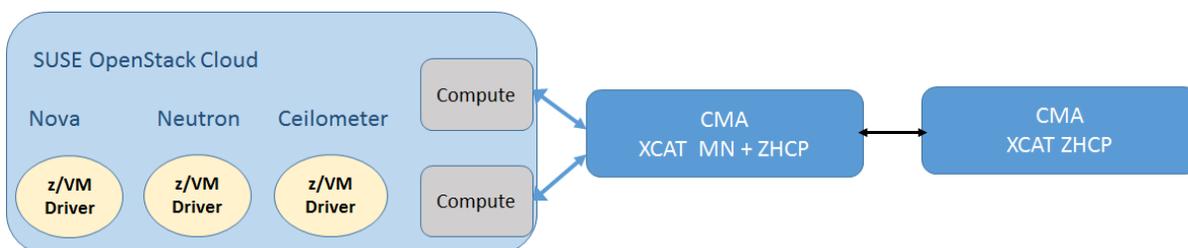
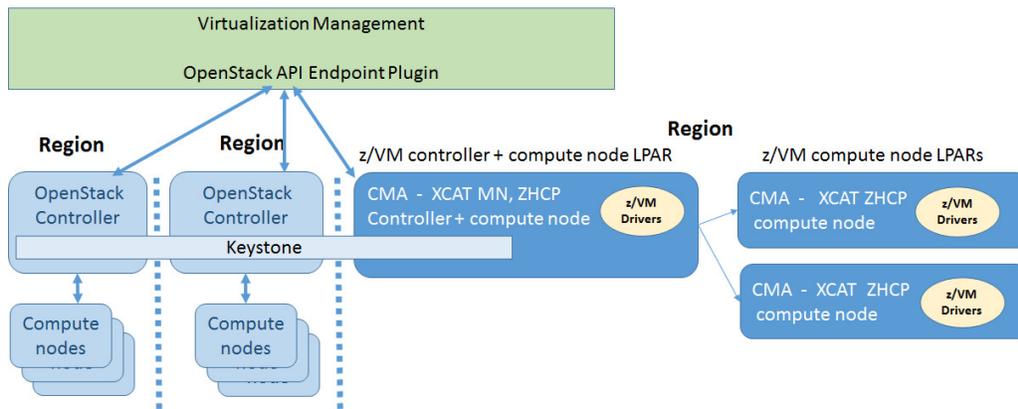**Cloud Management Context**

The CMA supports both manage-from and manage-to, with restrictions:

- Manage from requires one CMA to be in OpenStack controller mode, and it can only manage-to other z/VM LPARs:
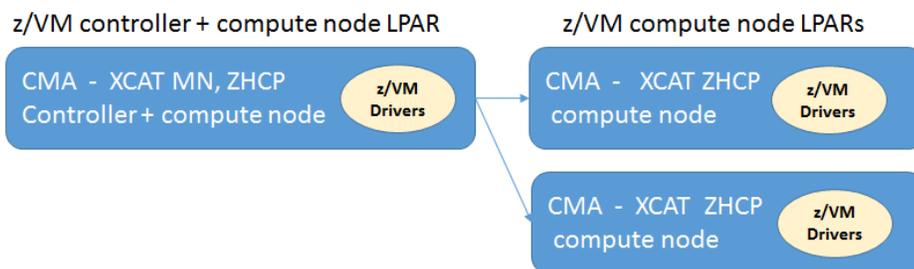
- Manage-to (using either OpenStack or VMware vRealize) requires OpenStack at the Liberty version or high (Newton for z/VM 6.4), and if the manage-from environment includes compute nodes targeting the CMA, the compute nodes must have the z/VM drivers installed. Example topologies:

With either the OpenStack or VMware vRealize environments, cloud management of the z/VM CMA is consistent with cloud management of other hypervisor environments.

z/VM Cloud Enablement APIs
NOTE: As this is a future technology that is not yet available, the capabilities described are subject to change.

The z/VM Cloud Enablement APIs are a future option that is currently planned to be generally available sometime in the first half of 2018. These APIs will provide a RESTful interface into z/VM that can be used by OpenStack and non-OpenStack environments to deploy Linux instances under z/VM.

The major differences between this and the z/VM CMA interface:

- The CMA is provided as an "appliance" guest that is part of z/VM. The enablement APIs are installed in a standard Linux guest provided by the client, using a supported Linux distribution of choice.
- OpenStack code will not be provided. Accessing this interface from OpenStack will require an OpenStack environment provided by a Linux distribution vendor (Red Hat, SUSE, or Canonical). In addition, the OpenStack compute node using the APIs must run on a Linux guest under z/VM.
- xCAT will be replaced by a "lighter weight" function to perform the similar actions of creating the Linux guest from an existing guest image.

The cloud enablement APIs are being developed as Open Source; code and documentations previews are available at:

- https://github.com/mfcloud/python-zvm-sdk
- http://cloudlib4zvm.readthedocs.io/en/latest/

Cloud Services Context

The following picture is the current high level architecture of how OpenStack and non-OpenStack environments would use the enablement APIs:
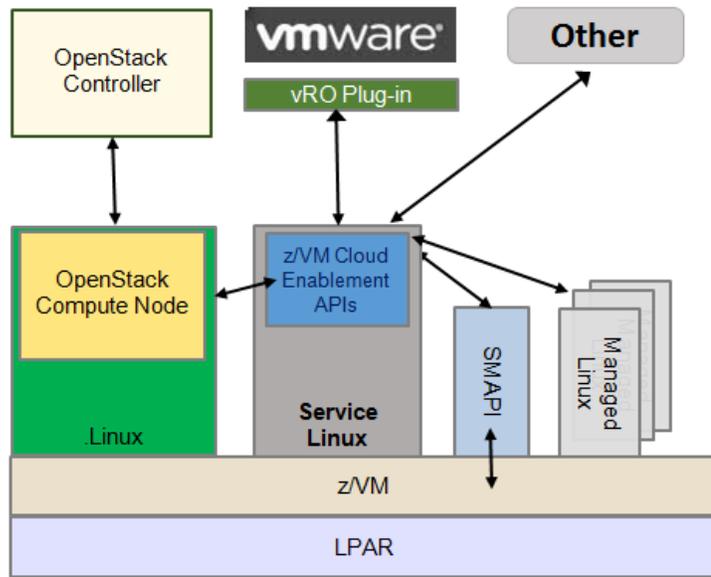
Cloud Services for OpenStack would be done through a compute node running on a Linux on z/VM guest, which would interface to the Cloud Enablement APIs. Non-OpenStack environments, such as VMware, would provide their own plug-in or other method to use the APIs.

The primary service available via this API would be IAAS. PAAS or SAAS would depend on the capabilities of the software using the interface. For example, the Heat component of OpenStack, or a VMware vRealize Orchestration (vRO) Plugin, could be customized to provide PAAS and/or SAAS functions.

Cloud Management Context

The manage-from capabilities when using the z/VM cloud enablement APIs will depend on the support the managing software has for running on Linux on z Systems.

KVM on IBM Z

Kernel Virtual Machine (KVM) on IBM Z refers to supported KVM environments provided by Linux distribution vendors. It does not refer to the IBM KVM for IBM z Systems product, which has been withdrawn and, for current users, has an end of service date of March 2018.

KVM is an open source hypervisor that is an extension of Linux, with a set of add-ons that allow virtual machines with their own operating systems and virtualized resources to run within a Linux guest. The KVM module is added to the Linux kernel to implement this virtualization.

KVM can be enabled in a Canonical Ubuntu 16.04 or later, or SUSE Linux Enterprise Server (SLES) 12.2 or later, running in an IBM Z or LinuxOne LPAR. It cannot be enabled in a Linux guest running under z/VM. From a command and function standpoint, it is operated in the same way as KVM on other Linux hardware platforms.

More information on KVM on z can be found at the following links:

- KVM for IBM z Systems overview and related material:
  https://www.ibm.com/systems/z/solutions/virtualization/kvm/
- KVM on Z technical blog: http://kvmonz.blogspot.com/
- SUSE support for KVM on z: https://www.suse.com/products/systemz/features/support-kvm-ibm-zsystems/
- Ubuntu support for KVM on z: https://insights.ubuntu.com/2017/06/15/ibm-canonical-a-virtualization-and-cloud-computing-r-evolution/

Cloud Services Context

Many Cloud Orchestration products, including OpenStack and VMware, provide KVM support. The key is to determine which specific configurations are supported, and map that against local requirements.

For OpenStack, Ubuntu on z System is supported at the Mikata and later release levels. The OpenStack controller and compute node functions can be enabled in a LPAR running Ubuntu, and an Ubuntu LPAR with KVM enabled can be configured as a compute node, with Linux guests deployed within the LPAR. The Linux guests can be any version that is supported on the IBM Z platform. An example topology:
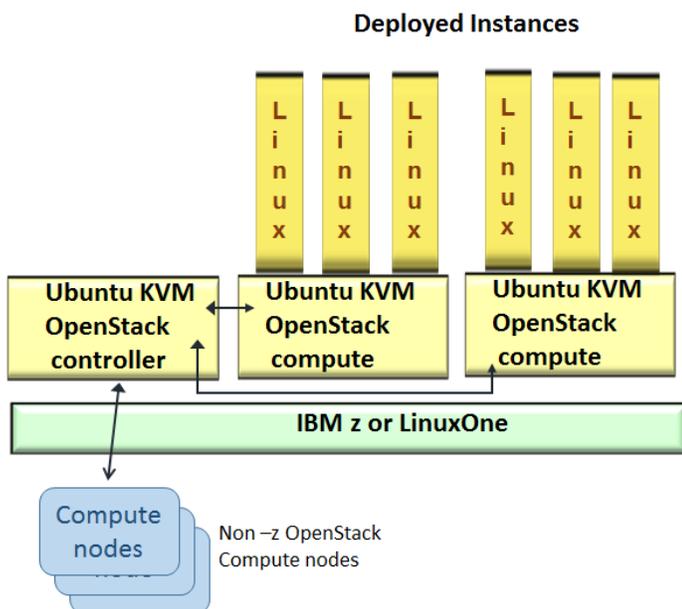
For VMware, the vRealize Automation OpenStack Endpoint plugin can connect to an OpenStack controller that is managing Ubuntu KVM on z compute nodes and deploy instances on those compute nodes.

Other products that use libvirt can connect to and manage an Ubuntu KVM LPAR to provide cloud services. libvirt is a toolkit for managing virtualization hosts, including KVM.

The primary service available via KVM would be IAAS. PAAS or SAAS would depend on the capabilities of the software that targets KVM. For example, the Heat component of OpenStack, or the VMware vRO OpenStack endpoint plugin, could be customized to provide PAAS and/or SAAS functions to instance deployed in a KVM for z LPAR.

Cloud Management Context

Since Ubuntu on z fully supports OpenStack, it can support an OpenStack controller running on IBM Z and be used as a manage-from platform to manage other Ubuntu KVM on z compute nodes (as well as non-z OpenStack compute nodes). The manage-from capabilities when using other options will depend on the support the managing software has for running on Linux on z Systems.

IBM Wave for z/VM

The primary usage of IBM Wave for z/VM is to provide a graphical user interface for managing z/VM and z/VM guests. One of the functions it provides is guest cloning, the ability to copy a source guest (either CMS or Linux) to one or more cloned instances.

More information on IBM Wave for z/VM can be found in its Knowledge Center at
https://www.ibm.com/support/knowledgecenter/en/SS6JTX/wave_welcome_v1r2.html

Cloud Services Context

Through its user interface, IBM Wave permits an administrator to quickly create new Linux guests by cloning from a defined prototype or another Linux guest. The interface is not meant for end users, but it well suited for a z/VM or Linux Administrator to quickly create Linux guests and manage their life cycle.

The IBM Wave cloning process supports IAAS level of cloud service, for Linux guests running under z/VM. PAAS and SAAS can be done but will require the heavy use of custom REXX EXEC and shell scripts.

Cloud Management Context

IBM Wave manages from z/VM, managing IBM Z and LinuxOne LPARs running z/VM. It does provide a REST API interface that would allow it to be managed to but other applications on or off the IBM Z and LinuxOne platform for some specific tasks against Linux guests. For example, the API does not support the cloning function, but does support the state management of already cloned instances.

Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. Kubernetes deploys containers as "pods" (one or more related containers) across one or more "worker nodes" (such as Linux instances running Docker) in a Kubernetes cluster. Pods can be assigned to logical "services" to allow access to the applications running in the pods from outside of the Kubernetes environment.

Command line and REST API interfaces are provided, to allow Kubernetes to be integrated with automated processes. A user interface (the Kubernetes dashboard) is available for users, with authorization and grouping (via "namespaces") for access to resources than can be deployed.

More information on Kubernetes can be found at https://kubernetes.io/, where documentation and code can be obtained.

Cloud Services Context

The following picture shows the high level architecture of Kubernetes:
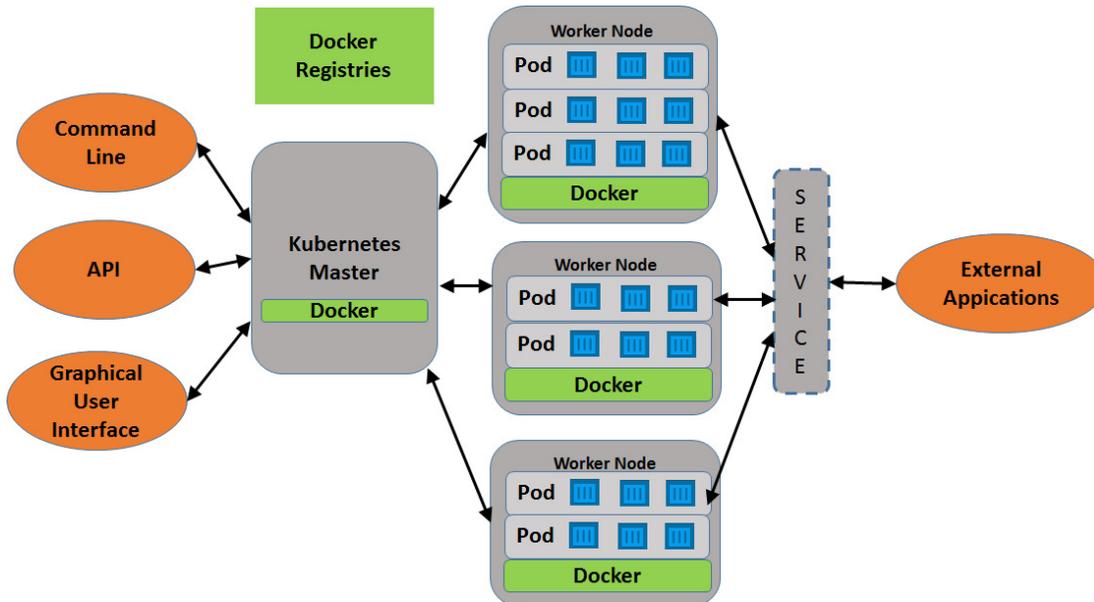
The Master node provides the management/user interface for orchestrating the environment. This is where deployments are created, which use Docker registries (public or private) to retrieve the required images. The worker nodes run the application containers in pods, which is a logical grouping of related containers. The pods can be scaled across nodes as needed or desired. Services are logical definitions that provide access to the applications within the pods from external applications. By replicating pods across nodes and front-ending them with service definitions provides high availability deployment of PAAS and SAAS applications in a Kubernetes environment.

Management Context

Kubernetes can be implemented as either a manage-from or manage-to solution for IBM Z and IBM LinuxOne. However, installation will require some detailed customization, particularly in creating Linux on z compatible Docker containers for some of the Kubernetes functions, such as the networking plugin.

IBM Cloud Private

IBM Cloud Private provides an infrastructure for developing and managing on premise container based applications. The major open source technologies that underlay IBM Cloud Private include:

- Docker, for containerized applications and functions
- Kubernetes, for managing a containerized environment

- Helm, the package manager for Kubernetes that allows detailed application definitions to be deployed from a repository.

On these technologies IBM Cloud Private adds additional capabilities such as:

- A private, secure repository for Docker images
- A management console for real time monitoring, logging, and event detection of the IBM Cloud Private infrastructure
- A graphical user interface to centralize activities for deploying, managing, monitoring, and scaling applications in this environment.
- A catalog in the graphical user interface which can deploy predefined applications via Helm charts (a set of catalog items containing IBM and open source applications is provided).

IBM Cloud Private has evolved from IBM Bluemix Local and IBM Spectrum Conductor for Containers to provide a private cloud environment for deploying and managing applications that run in Docker containers.

More information on IBM Cloud Private can be found at:

- IBM Cloud Private at IBM DeveloperWorks: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/W1559b1be149d_43b0_881e_9783f38faaff
- IBM Cloud Private Knowledge Center: https://www.ibm.com/support/knowledgecenter/en/SSBS6K/product_welcome_cloud_private.html

Cloud Services Context

The following picture shows the high level architecture of IBM Cloud Private:
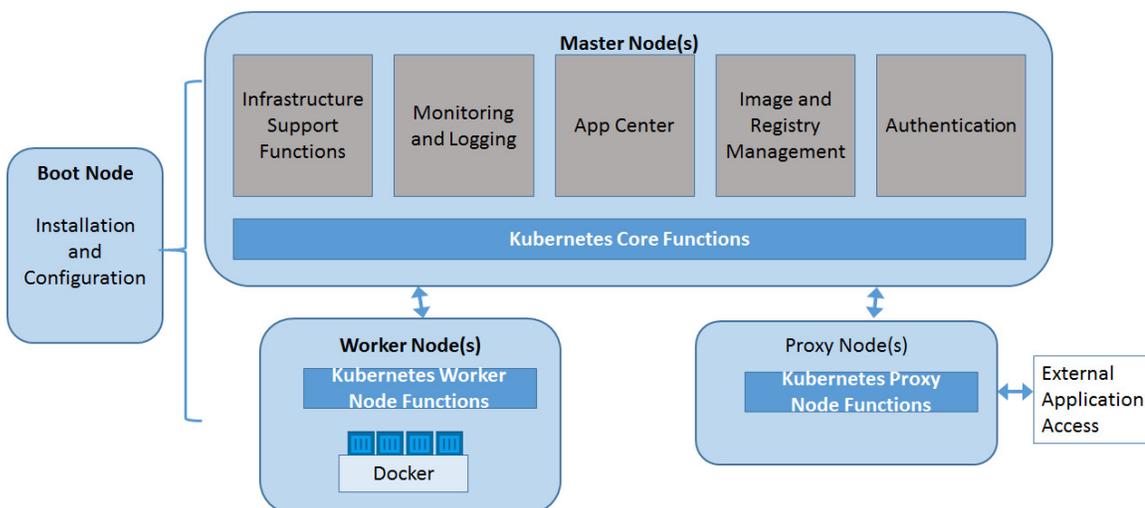
In IBM Cloud Private, the unit of deployment is the same as Kubernetes, a pod containing one or more Docker containers. They are deployed on worker nodes. For Linux on z Systems these are Linux LPARs or guests (Red Hat or Ubuntu) that have already been created from other methods (including the ones mentioned earlier in this paper). This enables PAAS and SAAS cloud services to be provided via the containers.

IBM Cloud Private bundles IBM Cloud Automation Manager. IBM Cloud Automation Manager integrates with cloud environments, including VMware and OpenStack based environments, to drive IAAS deployments to those environments, however at this time there is not official support for Linux on Z IAAS deployment.

Management Context

Currently IBM Cloud Private is a manage-to environment for IBM Z and IBM LinuxOne. Worker nodes are supported on Linux on z System instances, but the rest of the IBM Cloud Private infrastructure runs on Linux x86 (64 bit) or Linux on Power.

LinuxOne Community Cloud

The IBM LinuxOne Community Cloud is a public cloud environment for those wanting to test out Linux on z System distributions and applications in a LinuxOne sandbox environment, without impacting their current on premise infrastructure. It is supported by a partnership between IBM and Marist University. It provides a 120 day free trial of Linux running on z/VM to anyone who wants one.

More information on the IBM LinuxOne Community Cloud can be found at
https://developer.ibm.com/linuxone/

Cloud Services Context

The Linux Community Cloud provides a self-service interface that uses OpenStack API calls to the z/VM Cloud Manager Appliance to deploy Linux on z instances. The primary cloud service provided is IAAS, but some PAAS is planned.

Cloud Management Context

The Linux Community Cloud user portal runs on IBM LinuxOne, so the environment is a manage-from IBM LinuxOne to the LinuxOne LPARs supporting the deployed instances.

Potential Uses Cases for the Technologies

The following table summarizes potential use cases for each option. Note that the various options can complement each other based on local requirement. For example, DPM, KVM, or the z/VM CMA can be used to provide IAAS services on which IBM Cloud Private PAAS or SAAS services are run.

Cloud Option	Use Case
IBM Dynamic Partition Manager	Deployment of large servers that require an LPAR level amount of resources, and better performance than running under z/VM or KVM.

Cloud Option	Use Case
z/VM Cloud Manager Appliance	Standalone manage-from OpenStack environment, or integrated into an existing OpenStack or VMware environment for common deployment of IAAS and custom PAAS/SAAS services under z/VM
z/VM enablement APIs (future)	Integrated into existing OpenStack, VMware, or any desired automation or orchestration product for deployment of Linux on z instances under z/VM
KVM on IBM Z	Integration into an existing OpenStack, VMware, or libvirt-supporting automation or orchestration product for deployment of Linux on z instances under KVM
IBM Wave	Administrative deployment of Linux guests for end users to use
Kubernetes	Deployment of applications via Docker containers
IBM Cloud Private	Deployment of applications via Docker containers via a self-service catalog
Linux Community Cloud	Test drive and IBM Linux instance and software without impacting the current private infrastructure

Regardless of the use case, overall monitoring of the target hypervisor (of which these cloud resources will like be a subset of their use) will still need to be carried out using products such as the IBM Infrastructure Suite for z/VM and Linux (IISz), or, for KVM, IBM Tivoli Monitoring for Virtual Environments.

Summary

There are a number of technology options for implementing a cloud environment for Linux on IBM Z and IBM LinuxOne. This paper has attempted to position them from a few key technology perspectives. This is a starting point. The most important step, in the author's opinion, is to determine what type of cloud services one wants to provide for Linux on z, and from that derive the specific requirements needed in regards to the target environment. The better these requirements are understood, the better one can progress to assessing these options at a more detailed level and determining the optimum configuration to address the requirements.

Acknowledgements

Special thanks to the following people who reviewed this document and provided input:

- Mike Sine, IBM Washington Systems Center