Expertise
and insight
for the future

Eetu Mastosalo

Multicloud Management

Metropolia University of Applied Sciences

Bachelor of Engineering

Information technology

Bachelor's Thesis

20 February 2020

Author Title Number of Pages Date	Eetu Mastosalo Multicloud management 26 pages + 4 appendices 20 March 2020
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	IoT and Cloud Computing
Instructors	Tapio Wikström, Senior Lecturer

The goal of this study was to find out what kind of value a multicloud management system brings to businesses and organizations and to summarize scattered information about multicloud management in a single easily readable document. Another goal of this study was to produce a guide on how OpenShift 4.2 and IBM Cloud Pak for Multicloud Management are installed on top of Vmware infrastructure.

The study was conducted by researching the IBM Cloud Pak for Multicloud Management from online sources and via interview with a senior IT architect who works for IBM Finland and has deep knowledge of multicloud management. As a technical part of the project, a multicloud management system was installed in the test laboratory of IBM Finland where the system was used to study the installation procedures and capabilities of OpenShift 4.2 and IBM Cloud Paks.

Based on the study it can be concluded that a proper multicloud management system brings real value to businesses and organizations that are working with multicloud environments in the form of better data security, better control over their systems and overall reduced cost of multicloud resources. This study also concluded that installing OpenShift 4.2 and IBM Cloud Pak for Multicloud Management on Vmware infrastructure is a straightforward process.

Finally, a guide was compiled, presenting the value of multicloud management and how it can be installed and used to prevent uncontrolled use of unmanaged resources, increased costs and cloud vendor lock-in.

Keywords	Multicloud management, IBM Cloud Pak, OpenShift

Tekijä Otsikko Sivumäärä Aika	Eetu Mastosalo Multicloud management 26 sivua + 4 liitettä 20.3.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Iot ja pilvipalvelut
Ohjaajat	Lehtori Tapio Wikström

Insinöörityön tavoitteena oli selvittää, minkälaista lisäarvoa monipilviympäristön hallintajärjestelmä tuo yrityksille ja organisaatioille, kerätä hajanaista tietoa yhdeksi tietopaketiksi sekä luoda OpenShift 4.2- ja IBM Cloud Pak for Multicloud Management -ohjelmiston asennusohje.

Insinöörityö toteutettiin etsimällä tietoa monipilviympäristöjen hallintajärjestelmistä keskittyen IBM Cloud Pak for Multicloud Management -ohjelmistoon etsien internetistä sähköistä materiaalia sekä haastattelemalla Senior IT arkkitehtia Jouko Ruuskasta, joka työskentelee IBM Finlandilla ja tuntee monipilvijärjestelmät hyvin. Insinöörityön teknisenä osana oli monipilviympäristön hallintajärjestelmän asentaminen IBM Finlandin testilaboratorioon sekä asennuksen dokumentointi- ja asennusohjeen luominen. Asennuksen tarkoituksena oli tutkia asennuksen haasteellisuutta sekä OpenShift 4.2- ja IBM Cloud Pak for Multicloud Management -ohjelmiston kyvykkyyksiä.

Insinöörityön tuloksena selvitettiin, että monipilviympäristön hallintajärjestelmä todella luo kustannussäästöjä sekä lisää tietoturvallisuutta yrityksien ja organisaatioiden monipilviympäristöissä. Insinöörityössä myös huomattiin, että OpenShift 4.2- ja IBM Cloud Pak for Multicloud Management -ohjelmiston asentaminen Vmwaren infrastruktuurin päälle on mutkaton prosessi.

Insinöörityön tuotoksena on yksinkertainen ohje monipilviympäristön hallintajärjestelmien tuomasta arvosta yrityksille ja organisaatioille sekä asennusohje kokonaisen hallintajärjestelmän asennukselle.

Avainsanat	Monipilviympäristö, hallintajärjestelmä, pilvipalvelut

Contents

List of Abbreviations

MCM	Multi Cloud Management
CAM	Cloud Automation Manager
iCAM	IBM Cloud App Manager
ICP	IBM Cloud Pak
ICP4MCM	IBM Cloud Pak for Multicloud Management
POWERPC	Performance Optimization With Enhanced RISC Performance Computing
OCP	OpenShift Container Platform
RHCOS	Red Hat Core Operating System
DevSecOps	Development Security Operations
SDN	Software Defined Network
NSX-T	Network and security virtualization platform

1 Introduction

In the modern rapidly changing information technology landscape, the ability to quickly respond to changes is vital. A survey conducted by IBM [1] in 2018 says that 85% of companies are operating in a multicloud environment today. 98% of the surveyed organizations are planning to use multiple hybrid clouds within the next three years. Hybrid clouds can connect one or more public or private clouds to on-premises systems and can network multiple clouds together. Multicloud management aims to make utilizing multiple cloud vendors and hybrid clouds easy by establishing an environment where deploying and migrating applications across different clouds is easy and intuitive.

No cloud is the same. Different features, pricing, availability and security are key factors when determining what cloud provider to use. To get the best features and uptime from all clouds while also keeping the expenses at minimum requires centralized management and advanced automation. Two thirds of the surveyed organizations say an actively managed multicloud environment is crucial in reducing operating costs. [1.]

For this task, IBM has created a software solution called IBM Cloud Pak for Multicloud Management (ICP4MCM) which is part of IBM's Cloud Pak software lineup. ICP4MCM is a ready to use software solution that aims to tackle the challenges of multicloud.

The general aim of this thesis is to analyze the ICP4MCM solution and determine how the system is built and what business value it brings. The study also included writing a full installation guide to OpenShift 4.2 and IBM Cloud Pak for Multicloud Management that were installed in the test laboratory of IBM Finland in Munkkiniemi, Helsinki, as part of this thesis project.

2 Challenges of the multicloud environment

2.1 Current obstacles

Even though the overwhelming majority of enterprises are operating in a multicloud environment, relatively few grasp how to manage these environments. According to the IBM survey in 2018 [1], 98% of organizations are planning to use multiple hybrid clouds in the next three years but currently just 41% of them have a multicloud management strategy and only 38% have the abilities and tools to operate in a multicloud environment. Managing multicloud environments require the deployment and management of new virtual machines, storage, clusters, and other resources, and the maintenance of already existing and new resources. [2.]

Maintaining consistent policies and cluster life cycles across multiple platforms and scaling capacity based on demand to minimize cost becomes almost an impossible task when the complexity of the system gets larger. [2.]

One of the biggest obstacles in managing multicloud environments is security. According to IBM's 2019 Cost of a data breach report [3], the average cost of a data breach is 3,94 million dollars. Each cloud comes with its own tools and that increases complexity and cost. Making sure that every app in every cloud complies to security policies is increasingly difficult.

As different departments use their own clouds and applications there is a great risk of the cloud turning into shadow IT. Shadow IT is a term that describes IT resources that are not actively governed and because of that they usually are used very inefficiently and insecurely. [4]. This can cause unnecessary expenses and in the worst case security issues and possible data breaches.

2.2 Current solutions

The problems of multicloud can be overcome with DevSecOps, the centralized single pane of a glass control panel and automation and strict security policy compliance monitoring. Managing these separately and manually can easily become very complex and overwhelming as more clouds are added. That is where multicloud managers come into play. [2.]

2.2.1 Tackling complexity

A multi hybrid cloud can become very cluttered and hard to control as every cloud and service has its own control panel and tools that administrators need to be familiar with. IBM Cloud Pak for Multicloud Management tackles this challenge by combining the management of all clouds into a single pane of a glass view where administrators can control the whole cloud and on-premise applications from a single user interface with the most widely used tools. ICP4MCM cloud Overview screen can be seen in figure 1.

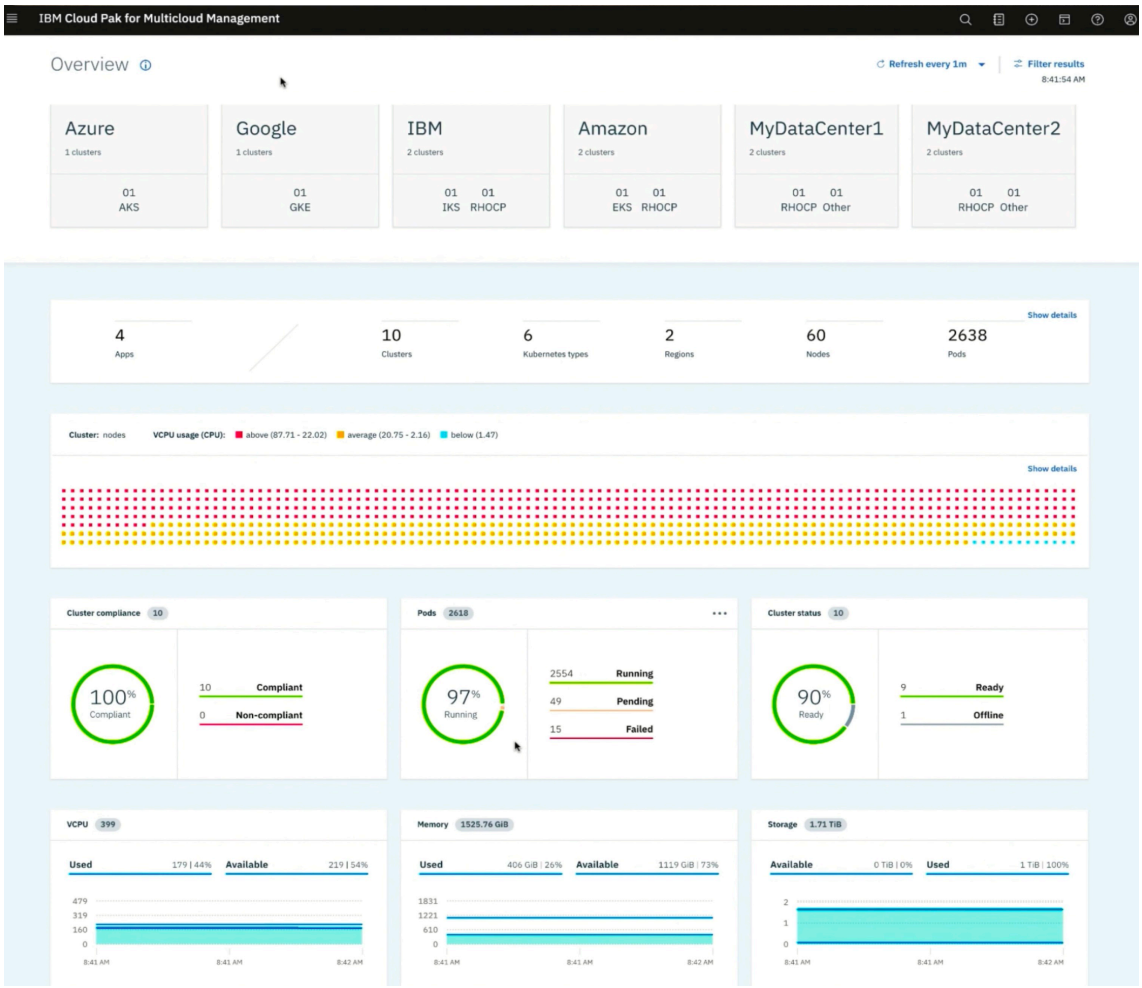

Figure 1. ICP4MCM Overview dashboard. Copied from IBM (2020) [18].

As can be seen from figure 1, vital information about clusters on different clouds and on premise data centers are visible on a single dashboard at a glance. IBM Cloud Pak for Multicloud Management gives organizations a single interface that offers open standard tools, self-service access to ready to use patterns and built-in governance. [2.] Automatic policy governance ensures that newly created resources such as Kubernetes clusters or virtual machines are policy compliant and secure without the need for manual error prone configuration. Self-service and ready-to-use patterns shift the repetitive simple work-loads away from administrators by allowing users to self-provision resources while still maintaining a consistent policy compliant service structure. [2.]

2.2.2 Enhancing security

ICP4MCM enforces an idea called Development Security Operations or DevSecOps where security is a part of the DevOps automation cycle right from the start. With the shorter and more frequent development cycles in today's cloud native world, security controls cannot be an afterthought. DevSecOps intertwines the goals and interests of development operations and security in all automations used. [2.]

Automated vulnerability scanning scans all resources for known vulnerabilities and outdated software and alerts administrators to take action. Periodical compliance checks can also be executed anytime from the start when resources are deployed.

3 Multicloud management competitors

IBM is not alone in the multicloud management market. There are multiple competing companies that each offer their own solution to tackle the problems that managing multicloud environments bring. All of the competitors offer their own advantages and disadvantages.

What distinguishes IBM Cloud Pak for Multicloud Management from the competition is that IBM's offering has all of the most used open source tools as well as the tools of IBM's partners for managing clouds, applications and security.

What combines Google Anthos, AWS Outposts and Azure Arc is that they all try to bind the customer to their own infrastructure. [3.] According to a survey that IBM conducted in 2018 [1], cloud vendor lock-in is one of the biggest problems of multicloud environments. In this matter, ICP4MCM is exceptional as it works on any cloud and vendor lock-in is not a threat [3].

The next sections describe some competitors to IBM Cloud Pak for Multicloud Management.

3.1 Google Anthos

Google Anthos is Google's offering in the multicloud management market. Google Anthos is a Kubernetes based application management platform that allows users to manage their Kubernetes clusters in cloud or on premise and deploy and manage their application lifecycles. [5.] Google Anthos has a unified user interface that can be seen in figure 2.

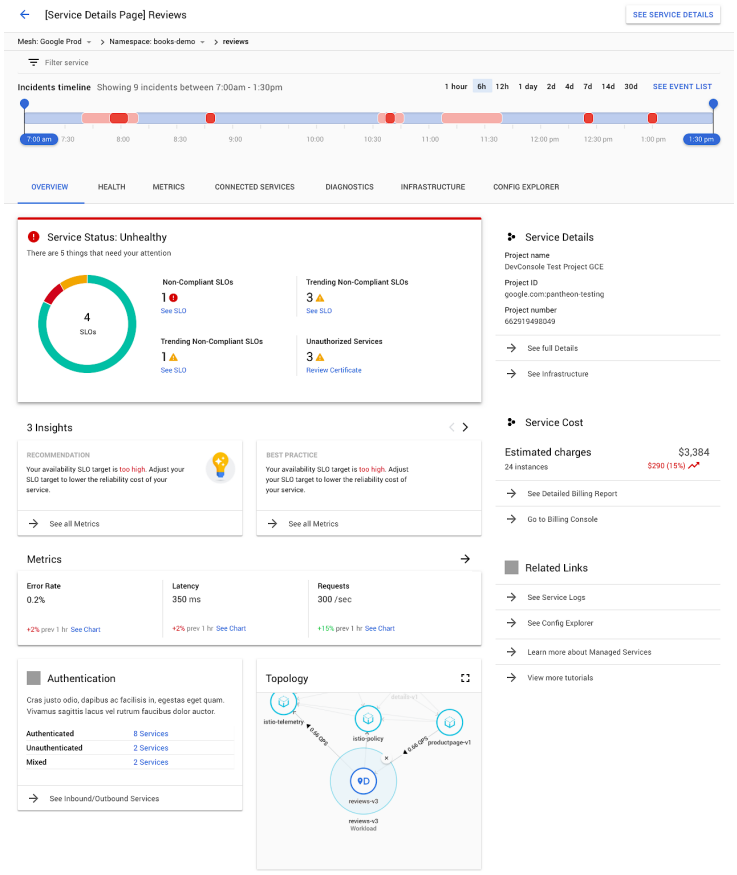

Figure 2. Google Anthos service status dashboard. Copied from Google (2020) [5].

As can be seen in figure 2, Anthos allows administrators to see the status of their clusters and deployments at a glance. The service mesh dashboard shows detailed status information of services and applications running in the connected clusters. Anthos provides centralized management and deployment of clusters that are policy compliant from the start. Anthos uses the configuration-as-code approach to enforce security that includes auditable and securable workflows, compliance enforcement and code free securing of microservices using Anthos service mesh, service mesh authorization and mesh certificate authority (CA). [5.]

3.2 Azure Arc

Microsoft's contribution to multicloud management market is their Azure Arc service that allows administrators to manage their virtual machines outside of the Azure cloud in other

clouds or on premise. This is done by installing an Azure Connected Machine agent on the connected machines. The agent connects the machine to Azure, gathers metrics and relays commands. [6.] The resource dashboard can be seen in the following figure 3.

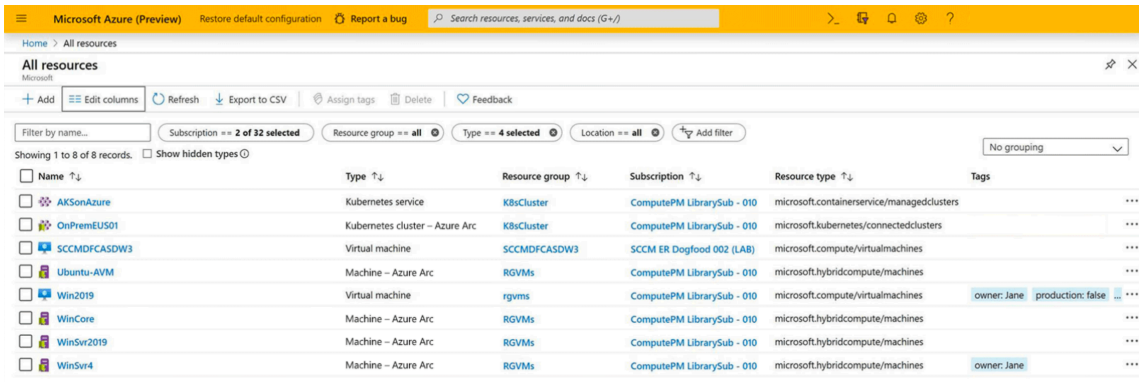

Figure 3. Azure Arc resource dashboard view. Copied from Microsoft (2020) [7].

As can be seen from figure 3, Azure Arc can manage physical machines, virtual machines and Kubernetes clusters on Azure, other clouds or on premise data centers. Azure Arc can show the status of machines and Kubernetes clusters everywhere on a centralized dashboard. It can also enforce Azure cloud security capabilities such as Azure Threat Detection, compliance enforcing and simplified auditing reporting. [7.]

3.3 AWS Outposts

Amazon Web Services also have their own solution. That is AWS Outposts which is somewhat different than other solutions, but the basic idea is the same. AWS Outposts is a fully managed service that extends the AWS infrastructure, services, tools and APIs to any data center or on-premise location. Unlike other multicloud management solutions described earlier, AWS Outposts brings the AWS cloud to on premises contrary to other competitors. With AWS Outposts, a customer buys a customizable physical server rack that is shipped to an on-premise location. There it will be connected to AWS and the on-premise data center. The AWS Outpost will be controlled from AWS web-portal just like any AWS service, but it will deliver on-premise latency and local data storage to the customer. It is also able to migrate workloads between the AWS cloud and the Outpost server. [8.]

4 IBM Cloud Pak for Multicloud Management

4.1 Architecture and major components

Thanks to the containerized structure of ICP4MCM, it can be provisioned anywhere where Red Hat OpenShift can run. The ICP4MCM reference architecture can be seen in figure 4.

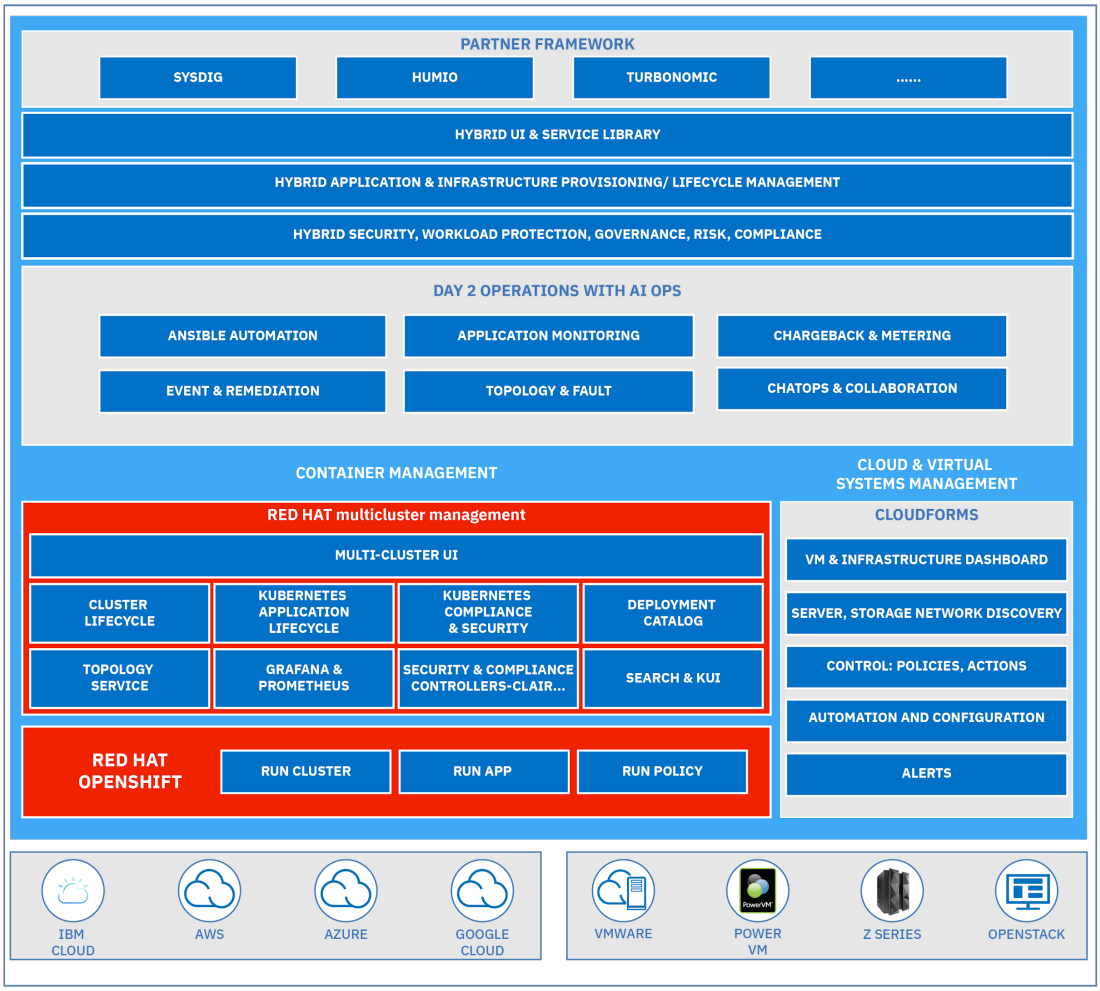

Figure 4. IBM Cloud Pak for Multicloud Management reference architecture. Copied from IBM (2020) [9].

ICP4MCM consists of multiple open source components as can be seen in the reference architecture in figure 4. ICP4MCM manages the cluster that it runs on and the connected clusters on any cloud, edge or on-premise location. Open source composition ensures transparency and enforces trust on the platform. The following are the six major components of ICP4MCM:

- Red Hat OpenShift Container Platform is a container orchestration platform that enables administrators to manage, monitor, scale, and secure containers similar to how they treat existing servers and virtual machines. It also provides easy access to persistent storage for applications or services, simple scheduling, load balancing, and automatic scaling by offering these critical functions and streamlined deployment to virtual or physical and public, private, or hybrid cloud. [9.]
- Red Hat CloudForms is a hybrid cloud lifecycle management tool that enforces compliance governance and service management. It gathers metrics data from clouds, clusters and applications and presents the data to administrators in an easily digestible form. CloudForms also delivers a chargeback system that allows organizations to monitor shared resources and charge the resource consuming party regardless of whether it is another organization, a customer or another department in the same company. Automated self-service simplifies, streamlines and accelerates resource usage for users but it shifts the control away from the IT department. For that reason, CloudForms has a request approval system that keeps the IT department in control. [9.]
- Ansible Tower securely automates resource allocation and application deployment. It logs all activity like who ran ansible, what changes they made, what the ansible playbook did and when and where it happened. Activity streams extend this by showing a complete audit trail of all changes made to Ansible Tower like job creation, inventory changes and credential storage. [9.]
- Multicluster management and event management are an enterprise grade Kubernetes multicluster management solution and an event management solution that allows visualizing the resources for easy management. Event Management can combine a considerable amount of information gathered from the clusters, show what events are happening in the clusters at a glance and assess problems if there are any. [9.]
- Hybrid security, Workload protection and Governance, Risk, and Compliance can automate compliance checking and vulnerability scanning. With this component, site reliability engineers can compile reports and run security compliance checks in the application release process to minimize security threats as early as in the development phase. [9.]
- The Partner framework allows bringing and integrating one's own favourite tools to ICP4MCM for easy transition to multicloud management. [9.]

4.2 ICP4MCM on ppc64le

ICP4MCM can run on multiple hardware architectures. Unlike most of its competitors, ICP4MCM supports 64-bit POWERPC little endian processor architecture in addition to the more traditional x86-64 architecture. Running selected workloads on POWERPC based servers has multiple benefits over x86-64. The latest Power9 processors offer up to two times the performance per core, 2.6 times more ram per socket and 1.8 times more memory bandwidth per socket than Xeon scalable processors [10]. Power9 was designed from the beginning with high performance parallel computing in mind like database and encryption processing. IBM's POWERPC architecture is also exceptional in running AI workloads thanks to its ability to integrate high bandwidth Nvidia nvlinks between GPUs and CPUs which is not possible on x86 hardware. The exceptional parallel processing capabilities can also be seen in the ability of the Power9 processor to run in simultaneous multithreading modes 4 and 8 whereas x86 is limited to simultaneous multithreading mode 2. This means that each Power9 core can run up to 8 threads simultaneously which enhances the processor throughput significantly. [11.]

4.3 Microservice architecture

Containers are replacing previously dominated virtual machines as the de facto standard when building and running applications. This new way of dissecting applications into small one function containers is called microservice architecture. Most containers today are run on Docker and orchestrated with Kubernetes but in an enterprise setting the focus is shifting away from Docker into more purpose-built container runtimes like CRI-O which is a lightweight solution that uses fewer resources and has a much smaller attack surface than Docker. Figure 5 shows CRI-O reference architecture.

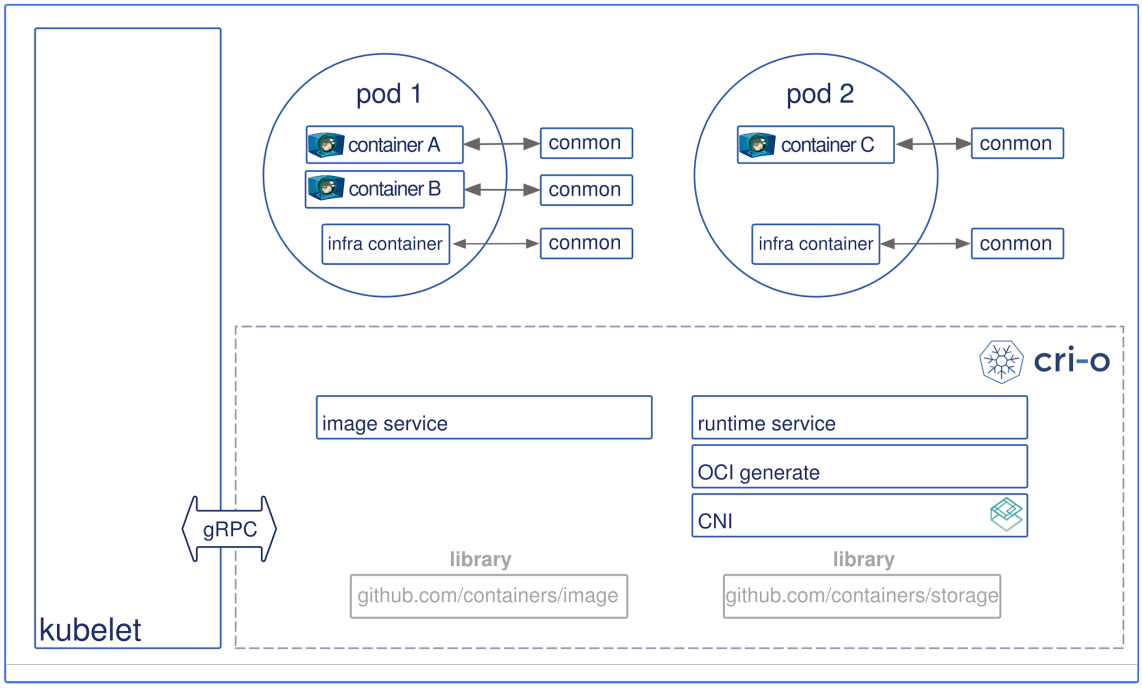

Figure 5. CRI-O reference architecture. Copied from cri-o (2020) [12].

As seen on figure 5, CRI-O is an implementation of the Kubernetes container runtime interface. Kubernetes contacts CRI-O via kubelet, and kubelet forwards the request to the CRI-O daemon to launch new pods and execute other functionalities. [12.] Today, more modern applications are built with microservice architecture in mind. This new architecture makes modern applications more resilient, reliable and easier to develop as different components can be independently developed, tested and deployed without affecting the function of the application as a whole. Even ICP4MCM is built on this new microservice architecture and runs on CRI-O runtime. The system orchestration is based on open source Kubernetes community direction. This allows easy integration of new tools into the system and detailed control and governance of individual components. [9.]

5 Installing Red Hat OpenShift 4.2

ICP4MCM runs on OpenShift so it is very important to carefully consider the hardware requirements of OpenShift and ICP4MCM before starting the installation. This section describes how to install a small standalone system that is good for testing and demonstrative purposes.

The first thing to consider is hardware capabilities. This installation of OpenShift with base ICP4MCM on top, will need a minimum of 16 CPU cores, 32 GB of memory and at least 560 GB of free storage space. Adding more components to ICP4MCM requires more resources. [13.] The following list describes the system requirements of Cloud Automation Manager and IBM Cloud App Management.

- The additional requirements of Cloud Automation Manager (CAM) are 12 CPU cores, 20 GB of memory when metrics are disabled and 65 GB of storage [13].
- The additional requirements of IBM Cloud App Management (iCAM) are two hosts, 12 CPU cores, 32 GB of memory and 100 GB of storage [13].

As can be seen from figure 6, the basic life cycle of OpenShift Container Platform is very simple. The lifecycle consists of creating the cluster, managing the cluster, developing and deploying applications and scaling the applications based on demand [14].

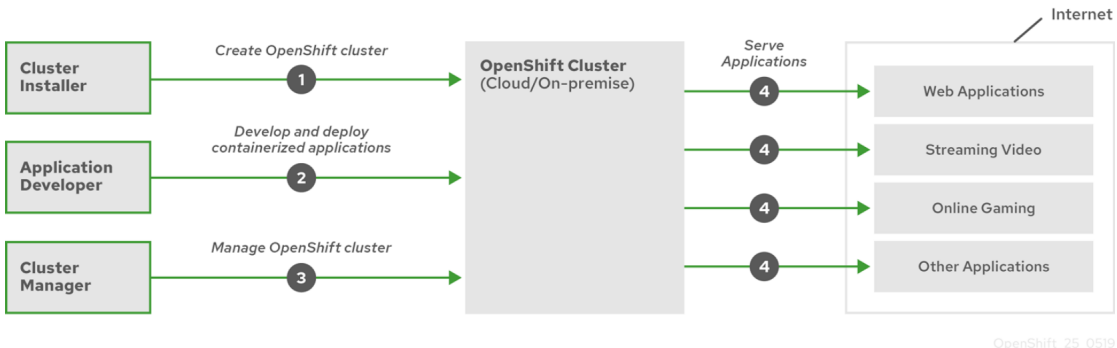

Figure 6. Overview of High level OpenShift Container Platform. Copied from Red Hat (2020) [14].

OCP can detect increased demand for different components from the collected pod metrics and scale pods horizontally up or down as is needed. Load balancers divide the traffic equally along the scaled pods. [15.]

When the hardware requirements are confirmed, it is time to start the installation. This guide gives instructions on how to install OpenShift and ICP4MCM on Vmware infrastructure. At least vSphere 6.7 is recommended but vSphere 6.5 will work with the restriction that NSX-T Software Defined Network (SDN) is not supported and another SDN or storage provider needs to be used [16]. Installation on bare metal is very similar. Installation with or without internet access is different and this guide describes the installation of OpenShift with internet access. This installation was performed from a Red Hat Enterprise Linux (RHEL) PC and all commands were executed from there unless stated otherwise. For this installation a DNS server must already exist. If the installation environment does not have a configurable DNS server available, a DNS server application can be installed on the load balancer machine with the following command.

```
$ sudo yum install named -y
```

5.1 Setting up and configuring pre requirements

This installation of OpenShift 4.2 requires seven virtual machines. One bootstrap machine, three control plane nodes, two compute nodes and a load balancer machine. The hardware specifications were chosen with ICP4MCM installation in mind and they were enough to run small basic workloads. The view of the created machines from vSphere can be seen in figure 7.

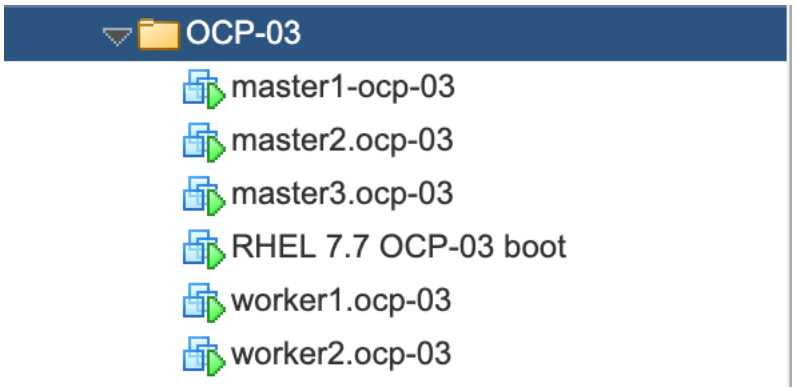

Figure 7. Screenshot of the created virtual machines on vSphere. Screenshot [16].

The bootstrap machine is used only for the installation and will be deleted afterwards. In figure 7, the bootstrap machine has already been deleted and the loadbalancer machine is named boot.

The hardware specifications of the machines are the following:

- Bootstrap: 8vCPU, 16GB Ram, 120GB Disk.
- Control plane node: 8vCPU, 24GB Ram, 120GB Disk.
- Compute node: 16 vCPU, 32GB Ram, 320GB Disk.
- Load balancer: 2vCPU, 4GB Ram, 60GB Disk.

The second step was to add Dynamic Name Service (DNS) records to the DNS server. As a result, the installer uses DNS records to find the right nodes. The needed records are as follows.

Kubernetes API, pointing to load balancer.

```
api.<cluster_name>.<base_domain>
api-int.<cluster_name>.<base_domain>
```

Routes, pointing to load balancer.

```
*.apps.<cluster_name>.<base_domain>
```

Etcd, pointing to control plane nodes.

```
etcd-0.<cluster_name>.<base_domain>
etcd-1.<cluster_name>.<base_domain>
etcd-2.<cluster_name>.<base_domain>
```

In addition, the following SRV records for all control plane nodes are needed.

Etcd.

```
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>.  86400 IN    SRV 0
10     2380 etcd-0.<cluster_name>.<base_domain>
```

After confirming that all DNS entries are set, all hostnames can be resolved successfully.

The next step is to generate the SSH key and add it to an ssh-agent. This enables the ability to perform installation debugging and disaster recovery on the cluster. This also enables passwordless SSH in the master and worker nodes when the installation is complete.

SSH keys can be generated with the following command.

```
$ ssh-keygen -t rsa -b 4096
```

The SSH-agent can be started as a background process with the following command.

```
$ eval "$(ssh-agent -s)"
```

The SSH key must be added to the SSH-agent with the following command..

```
$ ssh-add <path>/<ssh_private_key_file_name>
```

Next step was to start up the load balancer machine and install RHEL 7 or CentOS 7 on it. After installing and updating the operating system, load balancer software haproxy was installed. These commands were executed from the load balancer machine to install haproxy.

```
$ sudo yum install haproxy -y
```

Haproxy is configured by adding the required code (see appendix 1) to the end of haproxy configuration file haproxy.cfg. After the configuration, haproxy was restarted and enabled to start with the system by running the following commands.

```
$ sudo systemctl restart haproxy
$ sudo systemctl enable haproxy
```

5.2 Preparing the installation

Next the installation files were acquired by accessing the infrastructure page on the Red Hat OpenShift Cluster Manager site. The installation program for the correct operating system, Client tools and the installation pull secret as a txt file were downloaded. The pull secret allows the installer to authenticate and download installation files and images. Also RHCOS bios and iso files were downloaded.

The installer and client tools files needs to be extracted. The following commands were executed from the installation PC to extract the files.

```
$ tar xvf <openshift-install-linux.4.2.4>.tar.gz
$ tar xvf <openshift-client-linux.4.2.4>.tar.gz
```

The extracted client tools were moved into a folder that is in the PATH variable.

Next task was to create the installation directory to store the installer files with the following command.

```
$ mkdir <installation_directory>
```

The install-config.yaml file (see appendix 2) was copied to the installation PC and moved to the installation directory. The file must be indented correctly with spaces instead of tabs as the next task will fail if this file has any problems. The base domain, name and vsphere credentials and datastore names must be set into the install-config.yaml file. Also the pull secret and public ssh-key must be pasted to their respective places in the file.

On manual installations, compute node replicas must be set to 0. This is because the cluster cannot create and manage the compute nodes automatically on user provisioned manual installations. The compute nodes must be added to the cluster later.

The last step in preparing the installation was to create an HTTP server on the installation PC that the cluster can access. This HTTP server was used later to serve ignition files for the nodes. For an installation PC with Linux installed, the apache HTTP server software is more than adequate and can be installed with the following command.

```
$ sudo yum install httpd -y
```

5.3 Setting up the cluster infrastructure

In this part of the installation the infrastructure was prepared. The preparations include creating the Kubernetes manifest and ignition files and creating and configuring the Red Hat Core OS machines.

First the Kubernetes manifests were created by running the openshift-install script that came with the extracted installation program. The script was executed with the following command.

```
$ ./openshift-install create manifests --dir=<installation_directory>
```

The cluster-scheduler-02-config.yml Kubernetes manifest file in the manifests directory was modified to prevent Pods from being scheduled on the control plane machines by changing the “mastersSchedulable” parameter to false. If this was not done, it would have caused problems in the cluster later on.

Next task was to create the ignition files by running the following command.

```
$ ./openshift-install create ignition-configs
```

This command creates an auth directory with a kubeadmin-password file and kubeconfig file. The command also creates four files in the base directory. The files are called bootstrap.ign, master.ign, worker.ign and metadata.json. The ignition files must be moved to

the HTTP server. The ignition configuration files must be hosted because otherwise they are too large to fit in a vApp property and the installation will fail [17].

Next task is to configure the Red Hat Core OS (RHCOS) machines in vSphere. The RHCOS ISO file was uploaded into the vSphere datastore. In addition, a new folder in the vSphere data center was created to host the node virtual machines. The folder name must match the cluster name that was specified in the install-config.yaml file.

First, the bootstrap machine must be started by right clicking the virtual machine and pressing the open console button. In the console session, the tab must be pressed to open the configuration prompt and the configuration (see appendix 3) must be typed in one line. The details must be edited to fit the installation. To start the virtual machine installation, the enter key must be pressed. The virtual machine configuration must be repeated on all virtual machines except the load balancer.

5.4 Finishing the installation

When all virtual machines were turned on and configured, the bootstrapping process can be monitored by running the following command.

```
$ ./openshift-install wait-for bootstrap-complete --log-level=debug
```

The bootstrapping is complete when the message seen in figure 8 is displayed.

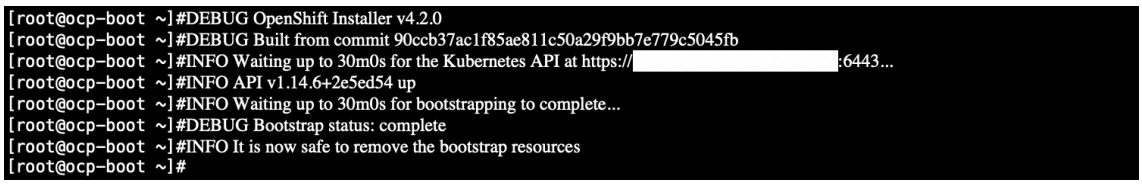

Figure 8. Screenshot of the OCP installer bootstrap complete message. Screenshot [18]

The bootstrap machine can be deleted from vSphere and removed from the load balancer after the bootstrapping is completed.

Next step was to log in to the cluster and verify that resources are recognized by running the following commands.

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig
$ oc get nodes
NAME       STATUS    ROLES    AGE  VERSION
master-01  Ready     master   63m  v1.14.6+c4799753c
master-02  Ready     master   63m  v1.14.6+c4799753c
master-03  Ready     master   64m  v1.14.6+c4799753c
worker-01  Ready  worker   76s  v1.14.6+c4799753c
worker-02  Ready  worker   70s  v1.14.6+c4799753c
```

When all nodes were confirmed to be visible, the installation can proceed.

The pending certificate signing requests (CSR) can be reviewed and all pending CSR's can be manually approved by running the following commands.

```
$ oc get csr
$ oc get csr -o go-template='{{range .items}}{{if not .sta-
tus}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs oc adm certificate ap-
prove
```

All cluster components must be online and running before finishing the installation. The component status can be monitored with the following command.

```
$ watch -n5 oc get clusteroperators
```

If the image-registry component is not available, a storage must be configured for it manually. For production clusters, the image-registry needs a persistent volume but in this installation an empty directory was used as storage for the registry. The image-registry storage was set to an empty directory with the following command.

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --
patch '{"spec":{"storage":{"emptyDir":{}}}}'
```

The installation was finished by running the following command and waiting for the install complete message.

```
$ ./openshift-install wait-for install-complete
INFO Waiting up to 30m0s for the cluster to initialize…
INFO Waiting up to 10m0s for the openshift-console route to be created…
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'ex-
port KUBECONFIG=/root/openshift4/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-con-
sole.apps.<cluster_name>.<base_domain>
INFO Login to the console with user: kubeadmin, password: STeaa-LjEB3-fjNzm-
2jUFA
```

The complete and detailed cluster status can be monitored with the following command.

```
$ oc get pods --all-namespaces
```

The final task was to log in to OpenShift by opening the presented web-console URL on a web browser and login with the given credentials. After a successful log in, the OpenShift 4.2 installation is ready and the installation of ICP4MCM can begin.

6 Installing IBM Cloud Pak for Multicloud Management

6.1 Preparing the installation

First, the ICP4MCM entitlement key must be purchased from IBM. To install ICP4MCM, the installation PC needs Docker to be installed. Docker was used to pull and deploy the ICP4MCM packages to OpenShift. Docker can be installed by running the following commands.

```
$ yum -y install  wget
$ wget https://download.docker.com/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo
$ sudo yum install docker-ce -y
```

The storage class, administrator password and the IBM entitlement key must be set to the config.yaml file (see appendix 4). It is then possible to log in to the OpenShift cluster with the following command.

```
$ oc login
username:
password:
```

The installation PC must be logged in to the public docker repository by using the IBM entitlement key. Logging in to the repository can be accomplished with the following command.

```
$ docker login cp.icr.io --username cp --password <your entitlement key>
```

Docker must then be used to pull the mcm-inception image to the installation PC with the following command.

```
$ docker pull cp.icr.io/cp/icp-foundation/mcm-inception:3.2.5
```

An installation directory must be created with the following command.

```
$ mkdir ibm-mcm
$ cd ibm-mcm
```

The following command must be executed to extract the installation file from the pulled image.

```
$ sudo docker run --rm -v $(pwd):/data:z -e LICENSE=accept --security-opt la-
bel:disable cp.icr.io/cp/icp-foundation/mcm-inception:3.2.5 cp -r cluster
/data
```

The kubeconfig file must be created with oc by running the following command.

```
$ oc config view > cluster/kubeconfig
$ cd cluster
```

6.2 Finishing the installation

After the installation has been prepared, the installation can be started by running the following command.

```
$ sudo docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/clus-
ter:z -v /var/run:/var/run:z -v /etc/docker:/etc/docker:z --security-opt la-
bel:disable cp.icr.io/cp/icp-foundation/mcm-inception:3.2.5 install-with-
openshift
```

The installer can take up to 45 minutes to run and the installation complete message shown in figure 9 can be seen when the installer has finished successfully.

Figure 9. Screenshot of the ICP4MCM installation complete message. Screenshot [19]

In figure 10, the welcome page of IBM Cloud Pak for Multicloud management can be seen.

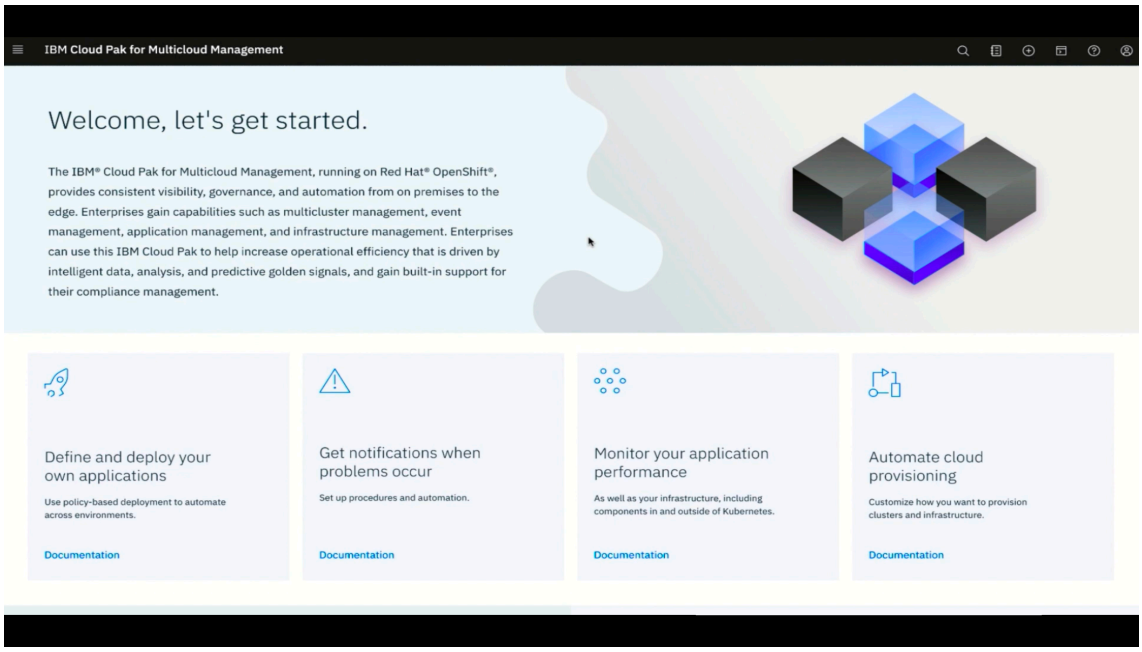

Figure 10. Screenshot of the ICP4MCM Welcome page. Screenshot [20].

As can be seen in figure 10, after the installation is complete, multiple additional packages can be deployed from the graphical web user interface including Cloud Automation Manager and IBM Cloud App Manager.

7 Business value

Multicloud has many benefits for businesses over a single cloud or an on-premise system. These benefits are hard to utilize without centralized management and control. ICP4MCM delivers the ability to centrally manage and control resources on the cloud, on the edge and on-premise. The greatest benefits are reduced cost, resilience and agility.

ICP4MCM gives businesses the benefit of lowering multicloud costs by giving them the ability to monitor and scale resource usage based on demand as well as transfer applications between clouds, for example, if the cloud provider changes their billing scheme. It also makes the work of administrators easier by giving them the ability to control everything from a single interface, which speeds up all cloud related processes reducing cost even further. [9.]

The benefit of resilience comes from ICP4MCM's ability to transfer applications on the fly from cloud to cloud and on-premise. If the organization's on-premise data center suffers a catastrophic failure, a backup workload can automatically start on the cloud and resume functioning there until the on-premise services are fixed. [21.]

ICP4MCM gives businesses the benefit of agility by allowing them to freely deploy their applications wherever they want in the world. Businesses can move their application from a data center in one country to another data center in another country with ease. This kind of maneuver may be needed, for example, if a country suddenly decides to withdraw from the European Union. Businesses can also move their workloads between data centers based on factors like local data laws or if another data center has faster connections and another cloud provider has better terms of service. [21.] Another important aspect of agility that ICP4MCM brings to businesses in the ability to scale applications beyond on-premise resources. Businesses may get a sudden need of more resources and scaling up your on-premise data center is slow and very expensive. ICP4MCM allows businesses to scale their applications to cloud and pay for only the resources that they use.

8 Conclusion

The goal of this bachelor's thesis was to explore the ICP4MCM solution, study the challenges that multicloud environments bring, study how ICP4MCM tackles these challenges, find out what value it brings to businesses and create an ICP4MCM system installation guide.

The goals were accomplished by studying the IBM Cloud Pak for Multicloud Management solution and installing a full multicloud manager system in the test laboratory of IBM Finland in Munkkiniemi, Helsinki. This study found out that a multicloud management system brings real value to businesses and organizations that are working with multicloud environments in the form of better data security, better control over their systems and overall reduced cost of multicloud resources. This study also concludes that installing OpenShift 4.2 and full ICP4MCM is a straightforward process which only requires some prior knowledge of Linux and containers in general.

This bachelor's thesis can be used, for example, as a reference when presenting ICP4MCM to companies and organizations as, for the time being, online information about ICP4MCM is very scarce and finding trusted sources other than blogs is difficult.

This bachelor's thesis could be supplemented by studying the whole IBM Cloud Pak lineup and compiling a full guide that includes all IBM Cloud Pak offerings. IBM Cloud Pak for data, integration, multicloud management, automation, applications and security all have their designated market segments in which they bring value, policy compliance and insight to companies and organizations.

References

1 How much would a data breach cost your business? 2019. Online. IBM. <https://www.ibm.com/security/data-breach> Accessed March 28, 2020.

2 Manage your multicloud environment with confidence. Online. IBM. <https://www.ibm.com/cloud/architecture/architectures/multicloudManagementArchitecture/overview> Accessed April 15, 2020.

3 Ruuskanen, Jouko. 2020. Senior IT Architect, IBM, Helsinki, Finland. Interview April 16, 2020.

4 Enhance and secure your hybrid, multicloud IT. Online. IBM. <https://www.ibm.com/cloud/smartpapers/MulticloudMgmt.pdf> Accessed April 16, 2020

5 Anthos technical overview. Online. Google. <https://cloud.google.com/anthos/docs/concepts/overview> Accessed March 28, 2020.

6 What is Azure Arc for servers. Online. Microsoft. <https://docs.microsoft.com/en-us/azure/azure-arc/servers/overview> Accessed March 28, 2020.

7 Azure Arc. Online. Microsoft. <https://azure.microsoft.com/en-us/services/azure-arc/> Accessed March 28, 2020.

8 AWS Outposts. Online. Amazon Web Services. <https://aws.amazon.com/outposts/> Accessed March 28, 2020.

9 Multi cloud management solution. Online. IBM. <https://www.ibm.com/cloud/architecture/architectures/multicloudManagementArchitecture/solutions> Accessed April 15, 2020.

10 Power9 servers overview. 2018. Online. IBM Power Systems. <https://www.ibm.com/downloads/cas/KDQRVQRR> Accessed March 28, 2020.

11 Jann, Joefon & Mackerras, P. & Ludden, John & Gschwind, Michael & Ouren, W. & Jacobs, S. & Veale, B. & Edelsohn, David. 2018. IBM POWER9 system software. IBM Journal of Research and Development; vol. 62, no. 4/5, paper 6. Online. <https://www.researchgate.net/profile/Michael_Gschwind/publica-

tion/325937212_IBM_POWER9_system_software/links/5be108ca92851c6b27aa24a6/IBM-POWER9-system-software.pdf> Accessed March 28, 2020.

12 Lightweight container runtime for Kubernetes. Online. Cloud Native Computing Foundation. <https://cri-o.io/> Accessed February 14, 2020.

13 Requirements. Online. IBM. <https://ocp42.cloudpak8s.io/mcm/cp4mcm_requirements/> Accessed March 20, 2020.

14 OpenShift Container Platform architecture. Online. Red Hat. <https://docs.openshift.com/container-platform/4.2/architecture/architecture.html> Accessed March 10, 2020.

15 Automatically scaling pods. Online. Red Hat. <https://docs.openshift.com/container-platform/4.2/nodes/pods/nodes-pods-autoscaling.html> Accessed March 20, 2020.

16 Vmware vCenter [Hypervisor]. Version 6.5.0. VMware Inc. January 20, 2020.

17 OpenShift Installer [computer program]. Version 4.2.0. Red Hat. January 21, 2020.

18 IBM Cloud Pak for Multicloud Management Installer [computer program]. Version 1.2. IBM. January 21, 2020.

19 IBM Cloud Pak for Multicloud Management [computer program]. Version 1.2. IBM January 21, 2020.

20 Installing cluster on vSphere. Online. Red Hat. <https://docs.openshift.com/container-platform/4.2/installing/installing_vsphere/installing-vsphere.html> Accessed February 18, 2020.

21 What is multicloud? Online. Red Hat. <https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud> Accessed February 14, 2020.

22 Cloud Pak for Multicloud Management. Online. IBM. <https://www.ibm.com/demos/collection/Cloud-Pak-for-Multicloud-Management/> Accessed March 10, 2020.

Haproxy.cfg

```
frontend openshift-api-server
      bind *:6443
      default_backend openshift-api-server
      mode tcp
      option tcplog
backend openshift-api-server
      balance source
      mode tcp
      server btstrap <bootstrap ip>:6443 check
      server master1 <master1 ip>:6443 check
      server master2 <master2 ip>:6443 check
      server master3 <master3 ip>:6443 check
frontend machine-config-server
      bind *:22623
      default_backend machine-config-server
      mode tcp
      option tcplog
backend machine-config-server
      balance source
      mode tcp
      server btstrap <bootstrap ip>:22623 check
      server master1 <master1 ip>:22623 check
      server master2 <master2 ip>:22623 check
      server master3 <master3 ip>:22623 check
frontend ingress-http
      bind *:80
      default_backend ingress-http
      mode tcp
      option tcplog
backend ingress-http
      balance source
      mode tcp
      server worker1 <worker1 ip>:80 check
      server worker2 <worker2 ip>:80 check
frontend ingress-https
      bind *:443
      default_backend ingress-https
      mode tcp
      option tcplog
backend ingress-https
      balance source
      mode tcp
      server worker1 <worker1 ip>:443 check
      server worker2 <worker2 ip>:443 check
```

install-config.yaml. Copied from Red Hat (2020) [14].

```
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: test
platform:
  vsphere:
    vcenter: your.vcenter.server
    username: username
    password: password
    data center: data center
    defaultDatastore: datastore
pullSecret: '{"auths": ...}'
sshKey: 'ssh-ed25519 AAAA...'
```

Virtual machine configuration

```
ip=<machine ip>::<default gateway>:<subnet mask>:<fully qualified domain
name>:ens192:none nameserver=<dns server ip> coreos.inst.install_dev=sda co-
reos.inst.image_url=http://<https server url>:<http server port>/<rhcos bios
file>.raw.gz coreos.inst.ignition_url=http://<http server url>:<http server port>/<ig-
nition file>.ign
```

config.yaml Copied from IBM (2020) [15].

```
# Licensed Materials - Property of IBM
# IBM Cloud private
# @ Copyright IBM Corp. 2019 All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or disclosure re-
stricted by GSA ADP Schedule Contract with IBM Corp.

---

# A list of OpenShift nodes
cluster_nodes:
  master:
    - worker-01
    - worker-02
  proxy:
    - worker-01
    - worker-02
  management:
    - worker-01
    - worker-02

# Storage Class
storage_class: <storage class available in OpenShift>

## You can set different storage class for logging.
## By default it will use the value of storage_class.
# elasticsearch_storage_class:

## If you are installing on ROKS environment please update following settings
## roks_enabled: set this to true
## roks_url: Openshift public service endpoint URL
## roks_user_prefix: User prefix used for users enabled on ROKS, Default value
is 'IAM#'.
## You can get users information from command 'oc get users'.
## ROKS settings
roks_enabled: false
roks_url: <roks_url>
roks_user_prefix: "IAM#"

default_admin_password: <your-password>
password_rules:
  - '(.*)'


## You can disable following services if they are not needed
management_services:
  # Common services
  iam-policy-controller: enabled
  metering: enabled
  licensing: disabled
  monitoring: enabled
  nginx-ingress: enabled
  common-web-ui: enabled
  catalog-ui: enabled
  mcm-kui: enabled
```

```
logging: disabled
audit-logging: disabled
system-healthcheck-service: disabled
multitenancy-enforcement: disabled

# mcm services
multicluster-hub: enabled
search: enabled
key-management: enabled
notary: disabled
cis-controller: disabled
vulnerability-advisor: disabled
mutation-advisor: disabled
sts: disabled
secret-encryption-policy-controller: disabled
image-security-enforcement: disabled
image_repo: cp.icr.io/cp/icp-foundation
private_registry_enabled: true
docker_username: cp
docker_password: <your entitlement key>
```