AIX Migration to Cloud with IBM Power Virtual Server

An IBM Systems Lab Services Tutorial

Aaron Bolding

Berjis Patel

Vess Natchev

ibmsls@us.ibm.com

TABLE OF CONTENTS

Chapter 1: Solution Overview

Introduction

A key question for AIX clients looking to leverage the benefits of IBM Power Virtual Server (PowerVS) is: how do I move my existing workloads to PowerVS? At the highest level, the process for most clients involves performing a save on-premise, transferring the save to PowerVS and performing a restore. For clients using Power Virtualization Center (PowerVC), the move to PowerVS involves using PowerVC to capture an Open Virtualization Appliance (OVA) image of the AIX workload and transferring the OVA to PowerVS.

Actually completing the migration process, however, involves many important details. AIX, PowerVS and networking skills are required. The goal of this tutorial is to provide step-by-step instructions for moving an AIX workload to PowerVS using three of the most popular methods.

Note that the focus of the tutorial is migrating the AIX operating system (OS) and disk/filesystem configuration. While the techniques described here can be used to move at least an initial save of a specific application, going into specific database or application requirements or synchronizing the data after a migration is beyond the scope of this document.

Use Cases

Migration via PowerVC OVA

In this case we will show how to migrate the entire logical partition (LPAR) and its associated disk(s) by using PowerVC's capability to capture the LPAR in an OVA. **Having a fully operational PowerVC**

environment is a prerequisite for this approach. Setting up PowerVC is beyond the scope of this tutorial.

Transfer System Backup Using the Public Internet

Here we will demonstrate how to install a mksysb backup from an on premises system to a Power Virtual Server Instance using the public network connection.

Transfer System Backup Using Cloud Object Storage

Lastly, we will show how to transfer your system backup to Cloud Object Storage and restore it into a Power Virtual Server Instance using a Linux Virtual Server Instance for staging.

Solution Components and Requirements

Components

Migration via PowerVC OVA

- *Fully functional PowerVC environment*
- *Sufficient space to capture an OVA image*
- *OVA created by PowerVC*

Transfer System Backup Using the Public Internet

- *AIX Virtual Server Instance with Public Internet interface*
- *Access from On Premise network to Public Internet*
- *AIX mksysb*

Transfer System Backup Using Cloud Object Storage

- *AIX Virtual Server Instance*
- *Direct Link Connect to IBM Cloud*

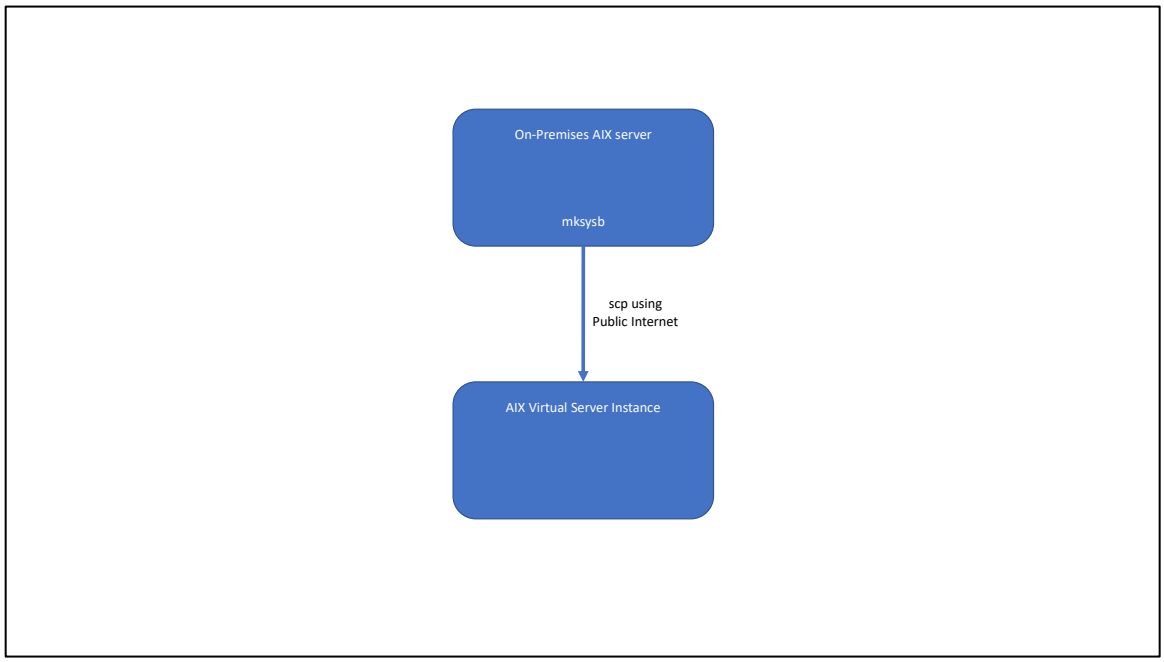
- *Linux Virtual Server Instance*
- *Cloud Object Storage Service*
- *s3fs-fuse Storage Driver for Linux*
- *NFS Server for Linux*
- *NFS Client for AIX*
- *mksysb for AIX*

Requirements

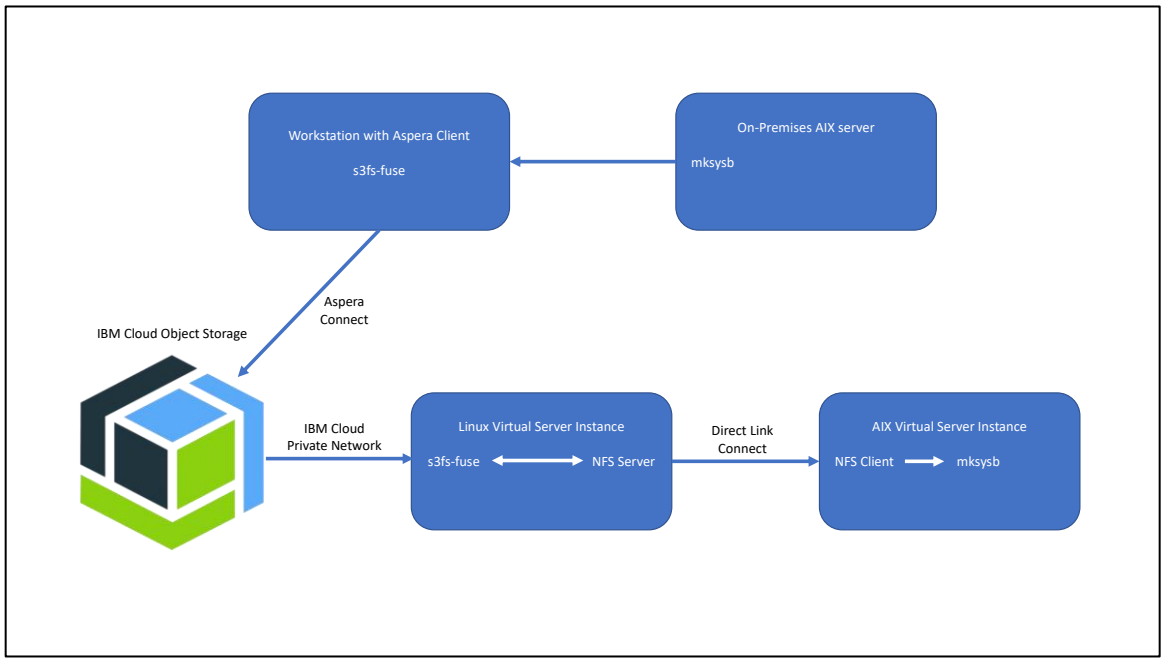
The system backup methods require a mksysb backup taken from the on-premises system to be migrated.

Solution Diagrams

Transfer System Backup Using the Public Internet



Transfer System Backup Using Cloud Object Storage



Chapter 2: Implementation

Migration via PowerVC OVA

Procedure to Configure IBM Cloud Object Storage

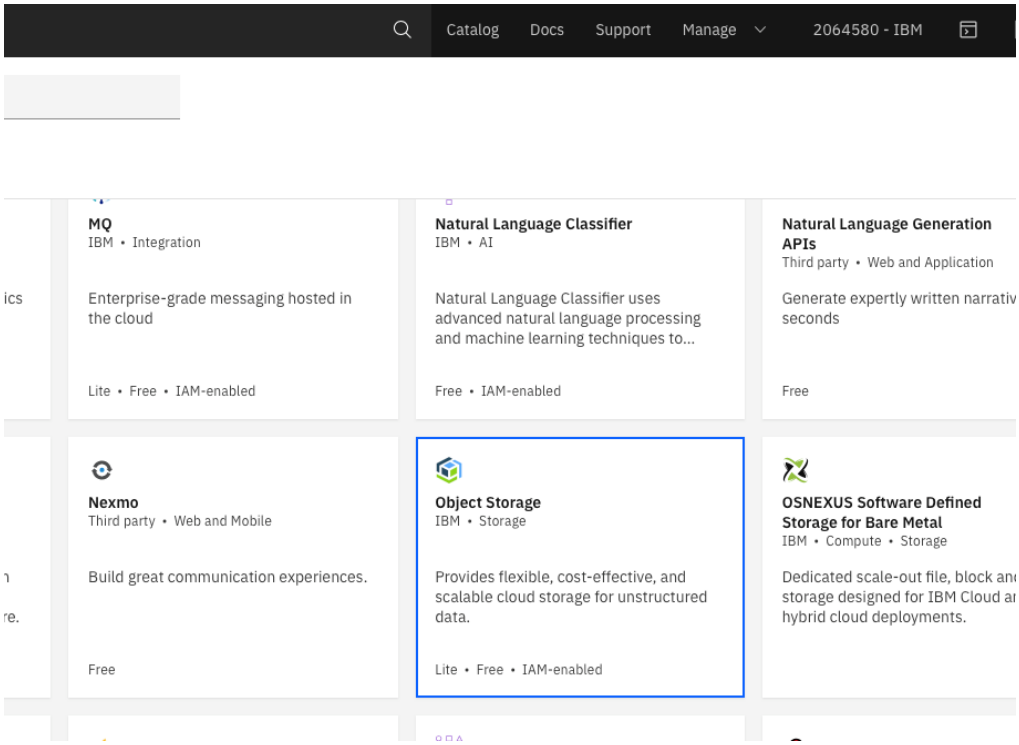
What is IBM Cloud Object Storage?

Cloud object storage is a format for storing unstructured data in the cloud. Object storage is considered a good fit for the cloud because it is elastic, flexible and it can more easily scale into multiple petabytes to support unlimited data growth. The architecture stores and manages data as objects compared to block storage, which handles data as blocks, and logical volumes and file storage which store data in hierarchical files.

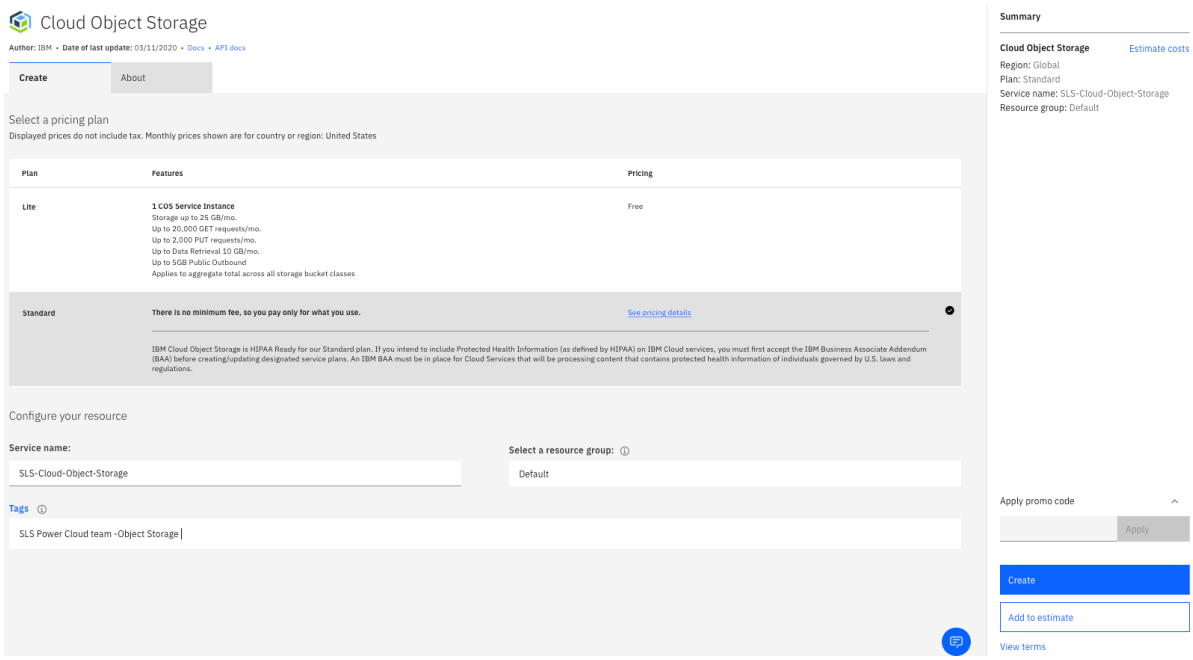
IBM Cloud™ Object Storage (ICOS) makes it possible to store practically limitless amounts of data, simply and cost effectively. It is commonly used for data archiving and backup; for web and mobile applications; and as scalable, persistent storage for analytics. Flexible storage class tiers with a policy-based archive let you effectively manage costs while meeting data access needs. The integrated IBM Aspera® high-speed data transfer option makes it easy to transfer data to and from IBM Cloud Object Storage, and query-in-place functionality allows you to run analytics directly on your data.

Creating an instance of IBM Cloud Object Storage in IBM Cloud

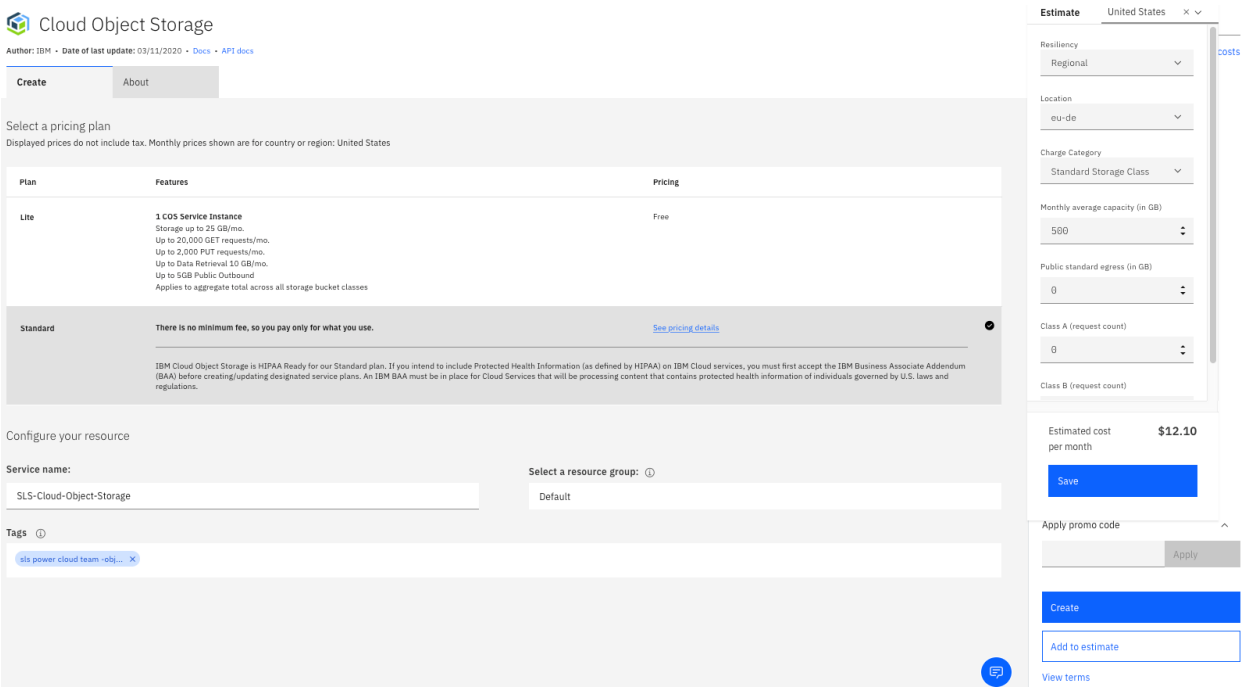
Login to the IBM cloud console, go to the Catalog and search for the Object Storage title and select it as below.



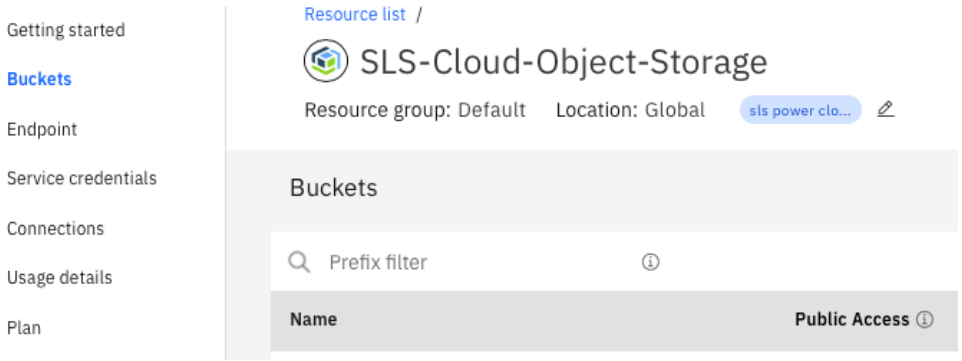
Give the service instance a name and choose either lite or standard plan. In our case we chose a Standard Plan and the instance name given was SLS-Cloud-Object-Storage.



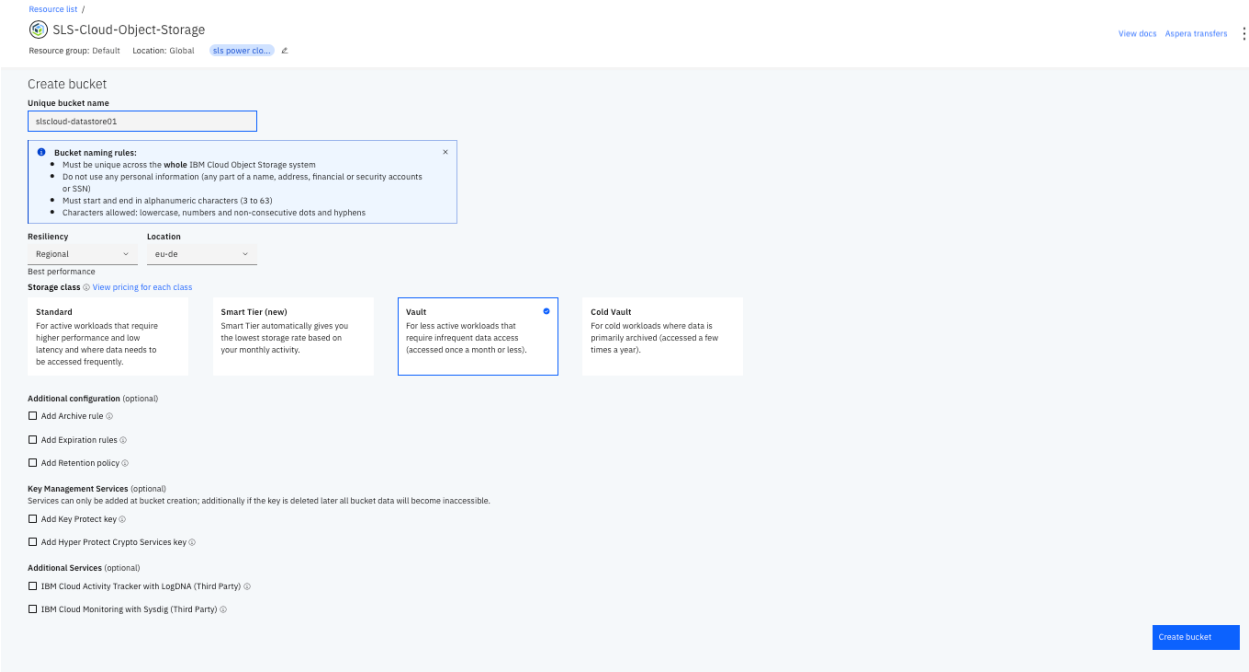
You can also check on the pricing depending on how much data you will need to store.



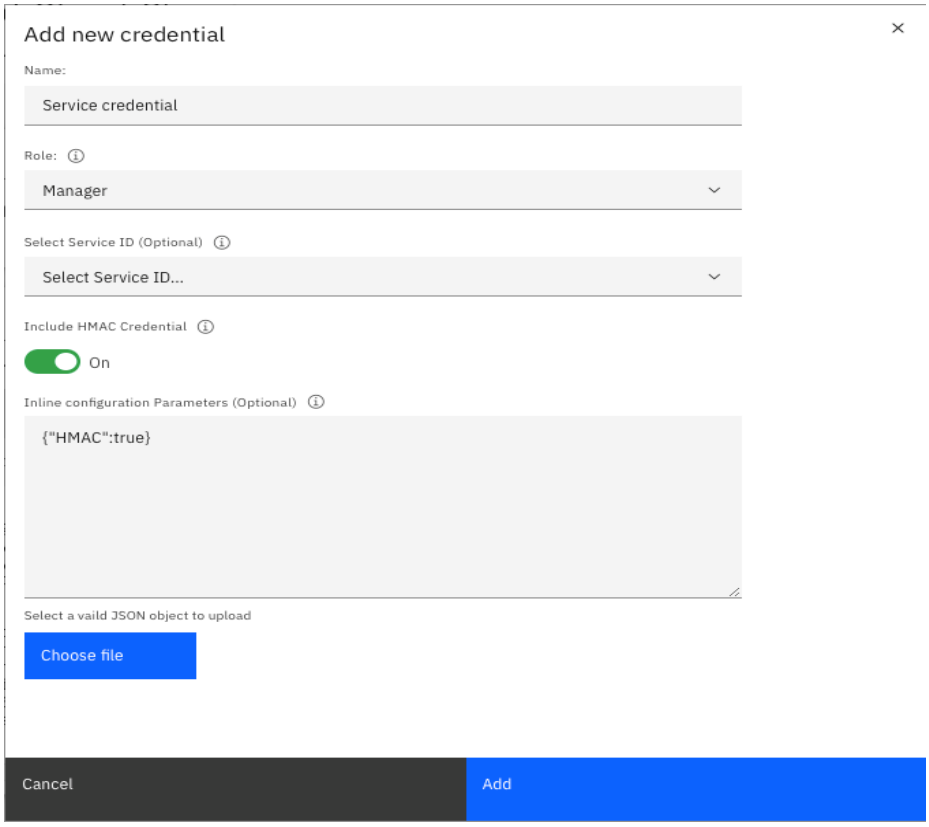
Click create and you will be automatically be redirected to your new instance.



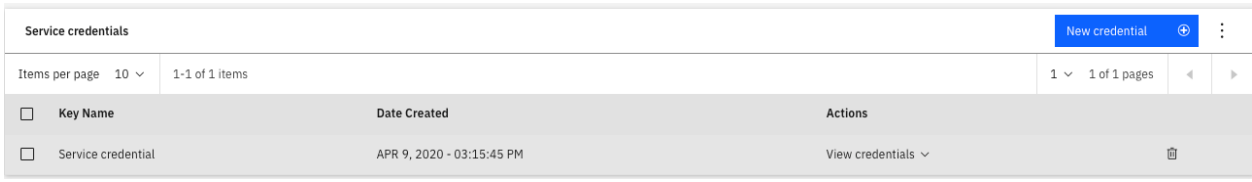
The next step is to create a bucket to store data, choose a name of the bucket with correct permissions.



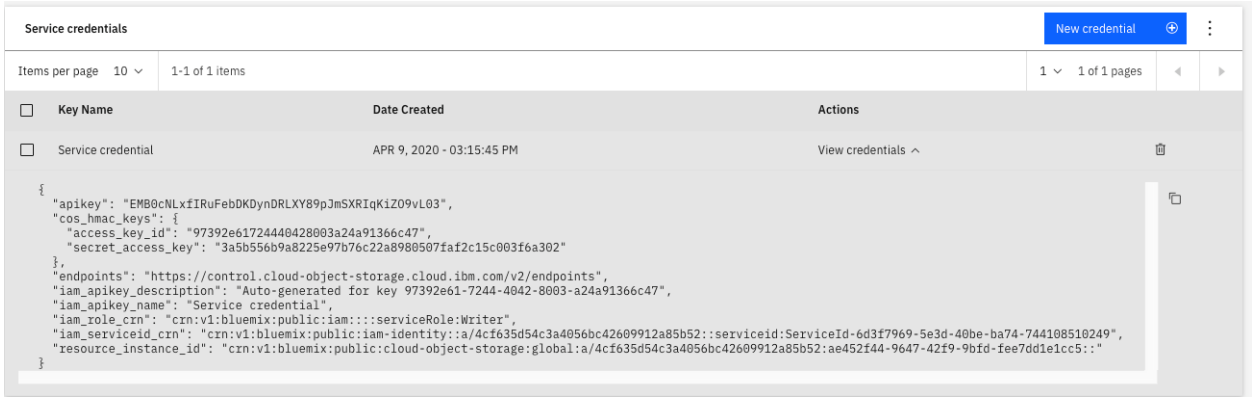
After creation of bucket we will need to create Service Credentials, which will provide necessary information to connect an application to Object storage.



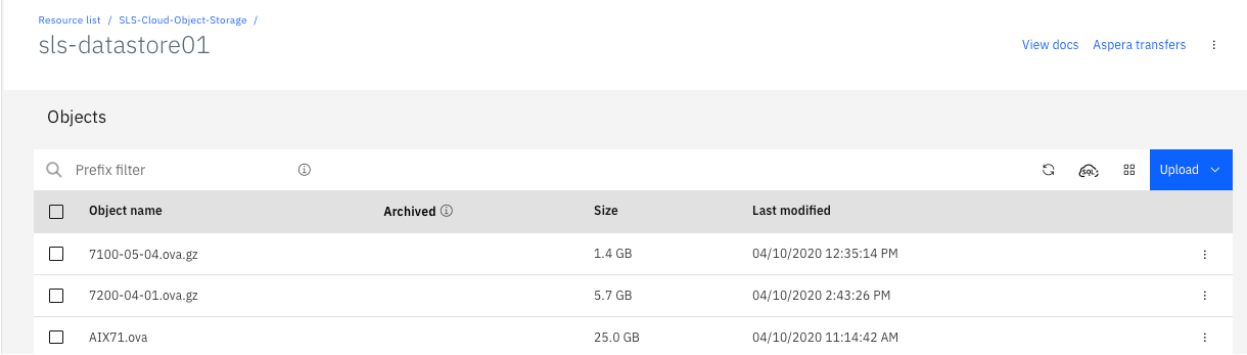
Below is the new service Credential which is created



After the service credential is created you can view credentials



We can now start uploading any ova file using the Aspera connect which is built in feature.



Note: You can either set Aspera as your default for any uploads. Link for more information on configuring Aspera high speed transfer in IBM Cloud.

https://cloud.ibm.com/docs/services/cloud-object-storage/iam?topic=cloud-object-storage-aspera

Note: You will need to install the IBM Aspera Connect client on your laptop. Attached is the link for the same.

https://downloads.asperasoft.com/connect2/?_ga=2.198106671.1294498643.1587399103-940898264.1586790978

Migrating an AIX Workload with PowerVC and OVA

We will now test the migration of an AIX VM to IBM cloud using export feature of PowerVC. For this we will need to make sure that the AIX VM which we are migration is managed by PowerVC.

Next step is to ssh to PowerVC VM and create an ova image. Link below to follow how to create ova image of a VM suing IBM PowerVC.

https://www.ibm.com/support/knowledgecenter/en/SSXK2N_1.4.4/com.ibm.powervc.standard.help.doc/powervc_export_image_hmc.html

Below is the command used to export the image from PowerVC. By default, the resulting OVA package is in the /var/opt/ibm/powervc/ova directory.

 # powervc-image export --image AIX71

Note: If you are facing issue with the above command please refer to PowerVC Knowledge Center.

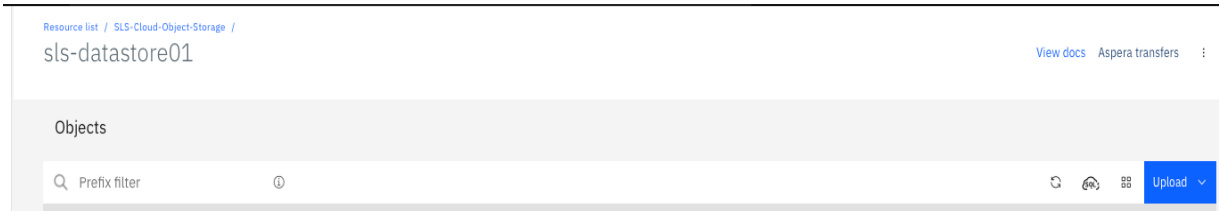
Next step is to transfer the PowerVC image to the Cloud Object Storage we created earlier. We will use the bucket **sls-datastore01** to store the ova file.

Note: As of writing this document there is no feature available to move the files from AIX or IBM i LPAR directly using IBM Aspera Fast Connect. For workaround we will transfer the ova file to local laptop (MacBook in this case) to store it to the Cloud Object Storage (COS). The IBM Aspera clients are also available for Widows and Linux.

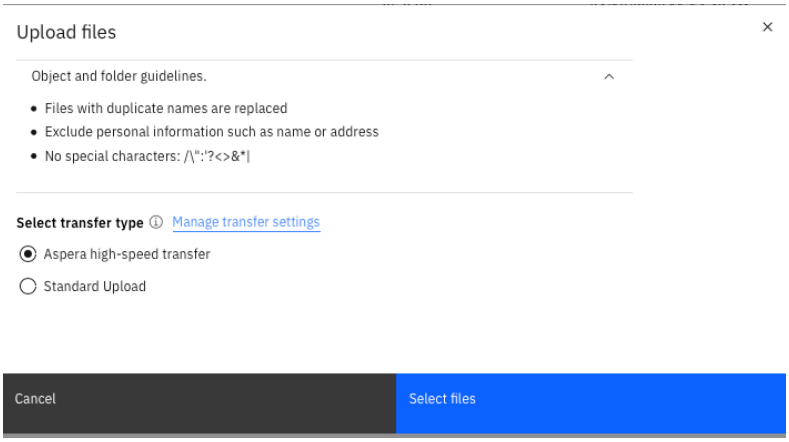
We will now install Aspera Client to the local laptop. Download the client from the link below and install the same on your laptop.

https://www.ibm.com/aspera/connect/

Gzip the ova file and move it from the PowerVC VM to your laptop. Next, start the Aspera connect from you IBM Cloud UI as shown below.

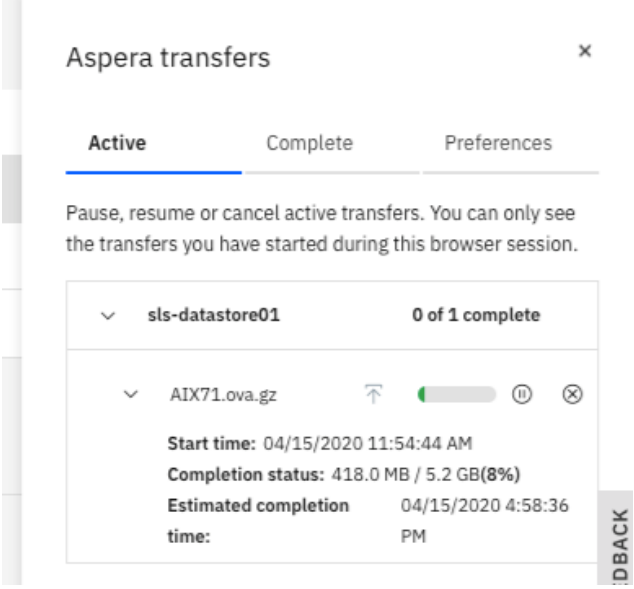


When you click on upload and select files you will see a pop-up window as below.



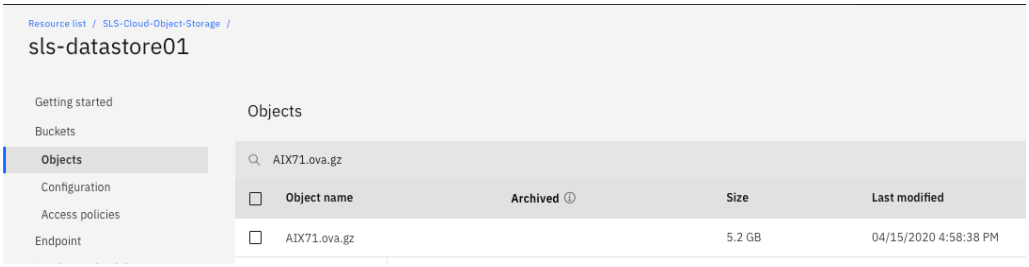
Note: Standard uploads can be maximum size of 200MB.

Click on “Select files” and it will browse to your laptop folder choose a file and start the upload. You can see the file transfer on you UI as below.

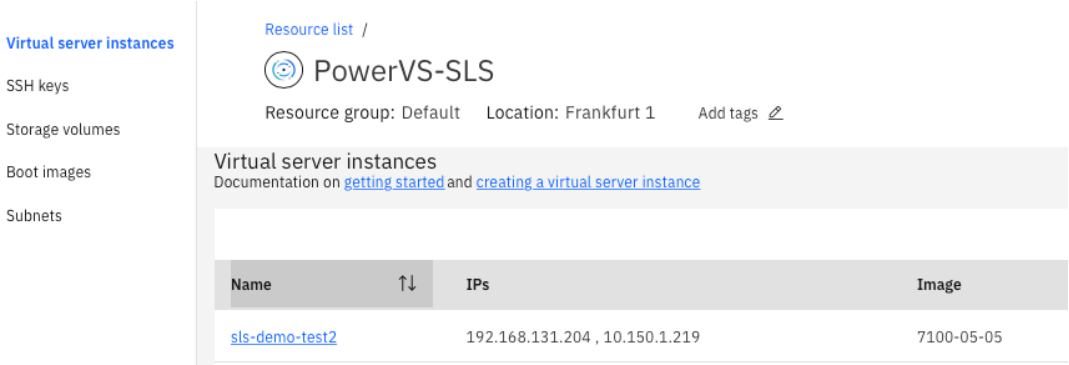


Note: It took about @4 minutes to upload a 5.2GB ova file

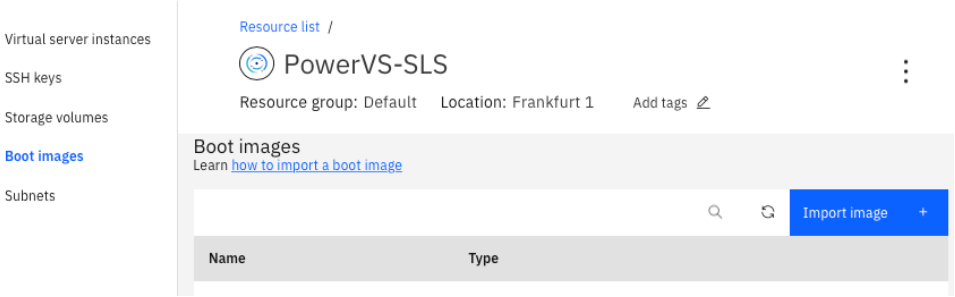
You can check the file in your bucket sls-datastore01.



Now we will need to import the ova file to our PowerVS instance. Go to Resource list → Services → PowerVS-SLS. You will be Virtual Server instance tab in the PowerVS-SLS service.



Select on boot images tab.



Before you import the image, we will need to have the below information.

Storage type (Tier 1 or Tier 3) being just an OS image we chose Tier 3
Region – eu-de

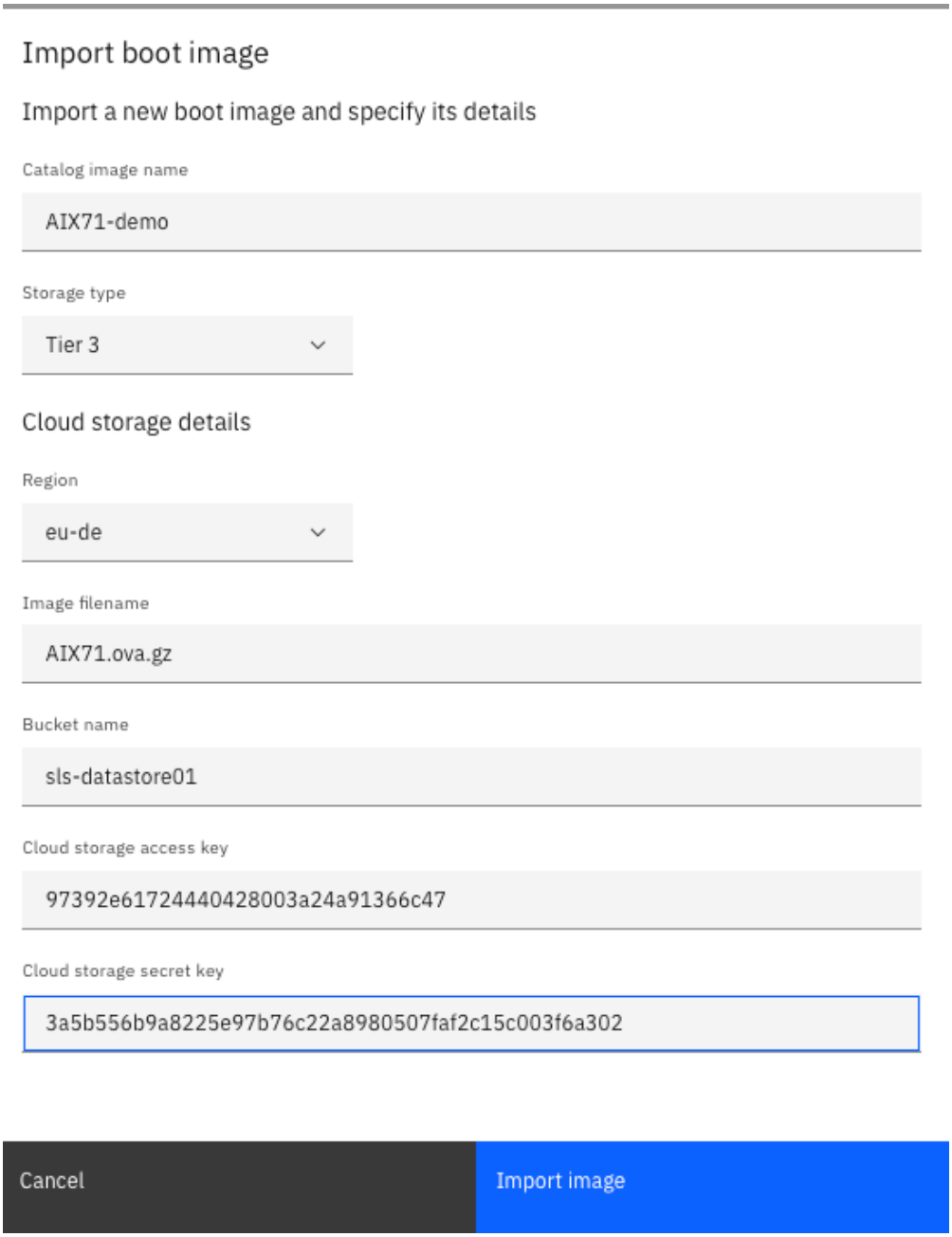
Image filename – AIX71.ova.gz selecting **Menu icon > Resource list > Storage > Cloud Storage Object**

Bucket name – sls-datastore01 select **Menu icon > Resource list > Storage > Cloud Storage Object name > Buckets**.

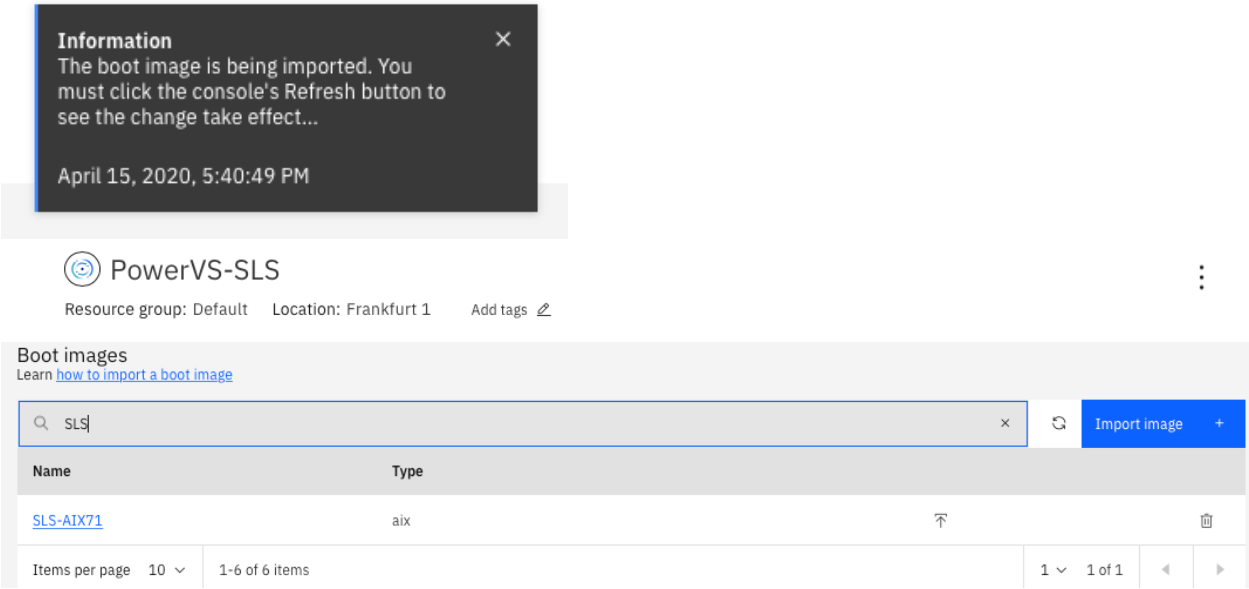
Cloud Object Storage access key -- select **Menu icon > Resource list > Storage > Cloud Storage Object name > Service credentials > View credentials**. Copy the access_key_id

Cloud Object Storage secret key -- select **Menu icon > Resource list > Storage > Cloud Storage Object name > Service credentials > View credentials**. Copy the secret_access_key

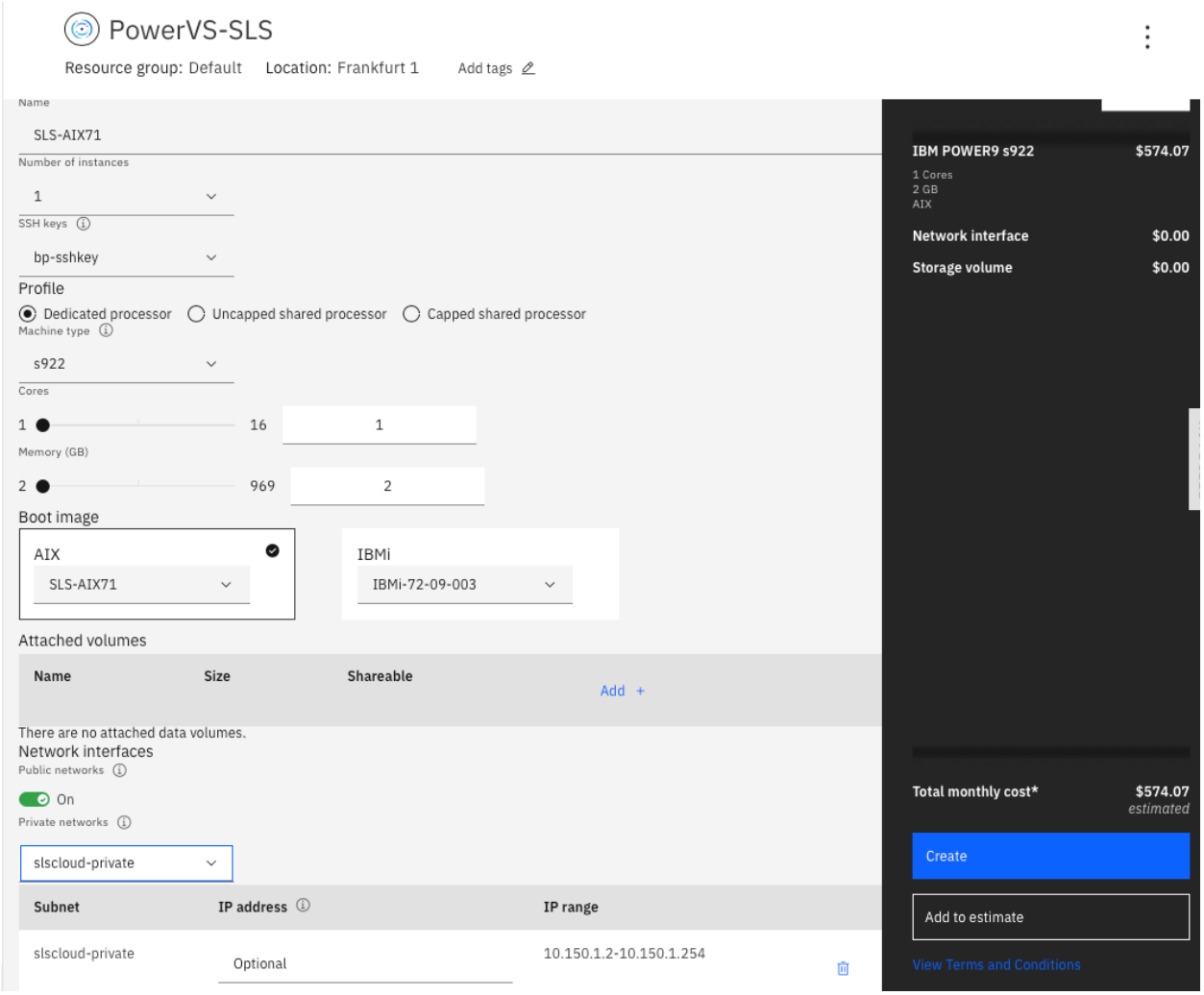
Now click on **"Import Image"** and you will then see a pop-up window to import a new boot image. Fill in the catalog-image-name, Storage type, Region, Image filename, bucket name and storage access key and secret key.



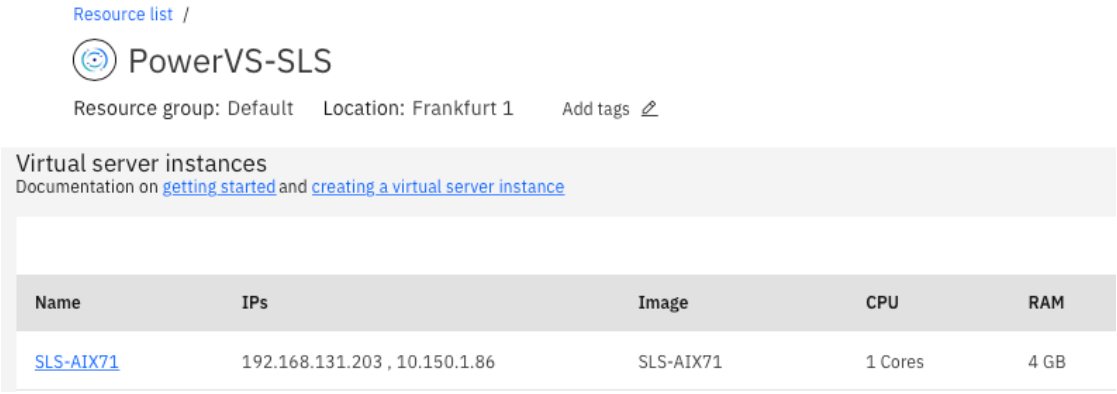
Verify the boot image is created



Once the image is imported, we will use this image to deploy a new AIX virtual Machine. Select Virtual server instance and click on create instance. Fill in the require Name and make sure you see the imported image which we just did in the above step.



You should now have a deployed LPAR /VM with the ova image as shown below.



Login to the LPAR and check for disk and system configs.

```
Berjis-MBP15:~ berjis$ ssh root@161.156.153.187
Last unsuccessful login: Wed May 29 13:34:16 CDT 2019 on ssh from infsvrlnx.ibmpowercloud.com
Last login: Wed Apr 15 21:38:35 CDT 2020 on /dev/pts/0 from 47.185.145.228
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
# lspv
hdisk0          00f7633dc62a92c9                    rootvg          active
#
# oslevel -r
7100-02
#
```

This concludes the PowerVC and OVA migration section.

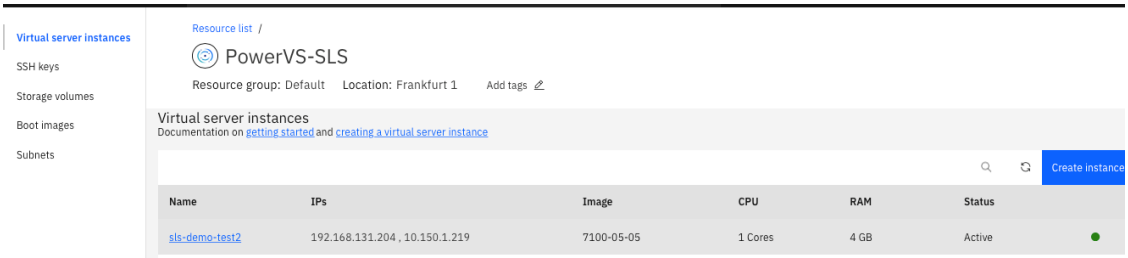
Transfer System Backup Using the Public Internet

Steps to Migrate AIX Virtual server using mksysb to IBM Cloud

1. *To move customer AIX VM to IBM cloud we will first need to create a Virtual Machine in IBM cloud using the stock image available. Login to IBM Cloud and follow the procedure described above to create a Virtual server.*
   Make sure we need to enable the public address during the creation of the VM in the cloud for us to have access, unless otherwise if we have VPN and Direct Link to the Power Systems colo.

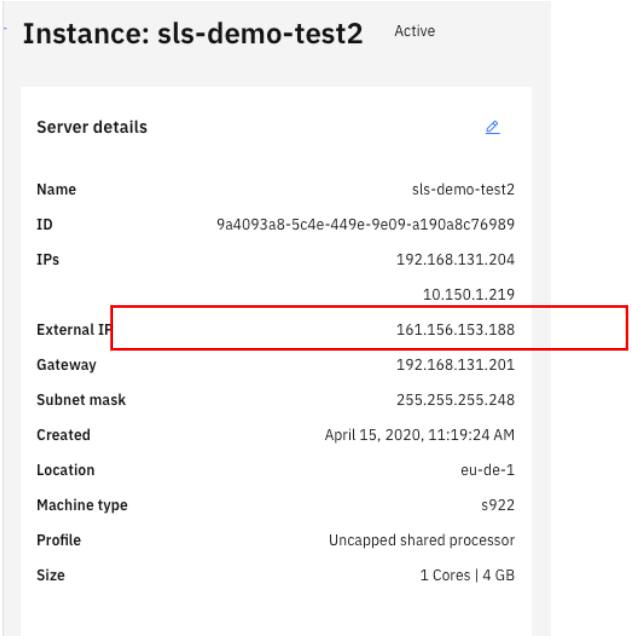
2. *We will use the current AIX Virtual Server in our example call sls-demo-test2 to show the migration. Shown below is the VM with internal and external IP Address. Check if you can reach the*

external IP address from your on-premise VM. Below is the existing VM which we will use for migration.



Note: Make sure when you create a VM you have enough space to fit the mksysb file.

3. *Note the External IP address which can be used to copy the mksysb*



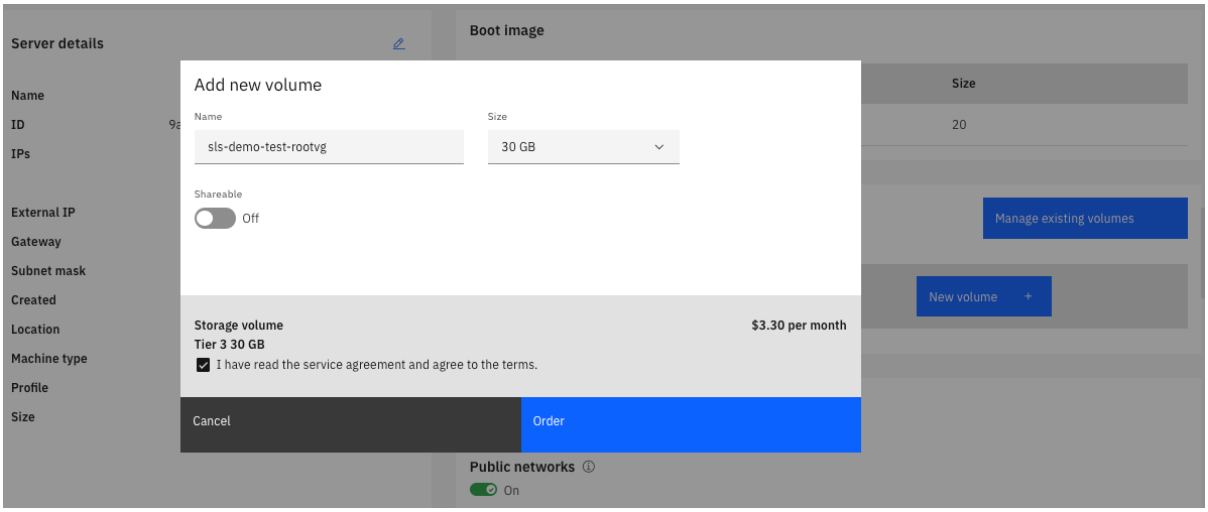
4. *Create a mksysb of the LPAR or VM which you will be migrating to the IBM Cloud.*
5. *Once mksysb is created you can now do a copy from the on-premise VM (or any location where your mksysb is residing) to the IBM Cloud VM using scp.*

Note: You can also gzip the mksysb to reduce time for the transfer.

Example showing copy to VM in IBM Cloud

```
[root@aix71-bp data]#scp aix71.sysb.gz root@161.156.153.188:/data
root@161.156.153.188's password:
aix71.sysb.gz                                          31% 1320MB   1.6MB/s   28:34 ETA
```

6. *Log back in the IBM Cloud virtual server (sls-demo-test2), and now we need to add additional disk, to restore the mksysb. Determine the size of disk required as per the original VM. In our case we added 30 GB of additional disk as below.*

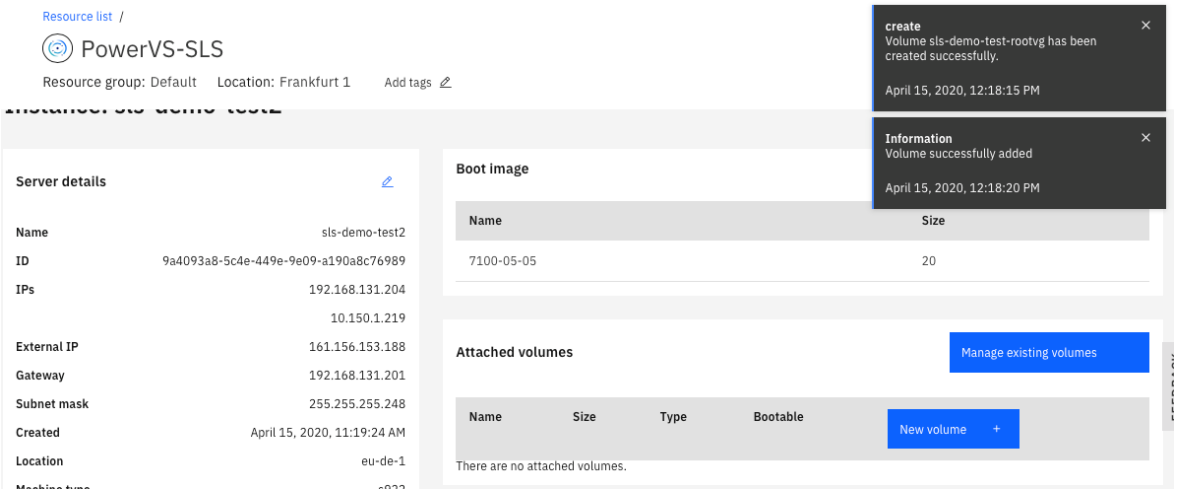


Note: If you are not sure what size of disk needs to be attached for the mksysb, run the below command on the mksysb file

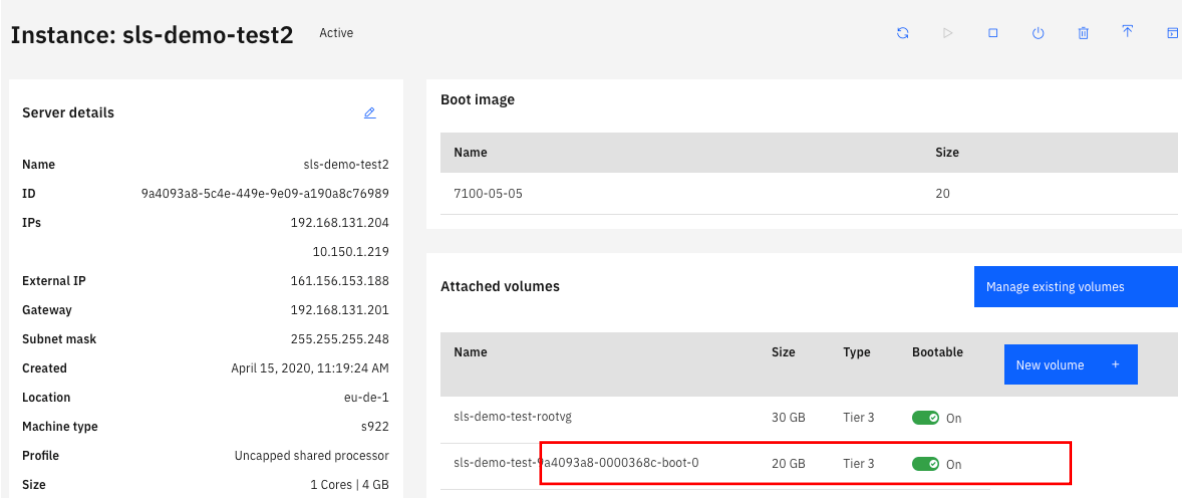
```
#restore -qf /data/aix71.sysb ./bosinst.data
x ./bosinst.data
#grep -p target_disk_data bosinst.data
target_disk_data:
        PVID = 00f7633dc62a92c9
  PHYSICAL_LOCATION = U8205.E6B.06633ER-V7-C6-T1-W500507680220AFD6-L0
        CONNECTION = fscsi3//500507680220afd6,0
        LOCATION = C6-T1-01
        SIZE_MB = 25600
```

HDISKNAME = hdisk0

7. *Below you will see the confirmation*



8. *You can check the new disk is attached by clicking the VM instance.*



Note: Make sure the disk has the bootable Flag ON.

9. *Now run cfgmgr on the VM and you should see both the disk*

```
[root@sls-demo-test2 /]#lspv
hdisk0          00f6db0a6c7aece5                    rootvg          active
hdisk1          none                                None
[root@sls-demo-test2 /]#
```

10. *We will now restore the mksysb using alternate disk mksysb (alt_disk_mksysb)*

Run the command alt_disk_mksysb with following option

 # alt_disk_mksysb -c <terminal device name> -d < the target disk to restore mksysb> -m <the full path of mksysb file>

-c -- to set up a terminal device during VM deployment. Without a valid terminal, the VM does not boot if it needs to open the terminal for any reason.

-d -- Specify the logical disk (hdisk) that is empty of a volume group label. In the example, the target disk is named hdisk1

-m -- Specify the mksysb full path archive that you transferred to the VM.

```
[root@sls-demo-test2 data]#alt_disk_mksysb -c /dev/vty0 -d hdisk1 -m /data/aix71.sysb
Restoring /image.data from mksysb image.
Checking disk sizes.
Creating cloned rootvg volume group and associated logical volumes.
Creating logical volume alt_hd5.
Creating logical volume alt_hd6.
Creating logical volume alt_hd8.
Creating logical volume alt_hd4.
Creating logical volume alt_hd2.
Creating logical volume alt_hd9var.
Creating logical volume alt_hd3.
Creating logical volume alt_hd1.
Creating logical volume alt_hd10opt.
Creating logical volume alt_hd11admin.
Creating logical volume alt_lg_dumplv.
Creating logical volume alt_livedump.
Creating /alt_inst/ file system.
Creating /alt_inst/admin file system.
Creating /alt_inst/home file system.
Creating /alt_inst/opt file system.
Creating /alt_inst/tmp file system.
Creating /alt_inst/usr file system.
Creating /alt_inst/var file system.
Creating /alt_inst/var/adm/ras/livedump file system.
Restoring mksysb image to alternate disk(s).
Linking to 64bit kernel.
Changing logical volume names in volume group descriptor area.
Fixing LV control blocks...
forced unmount of /alt_inst/var/adm/ras/livedump
forced unmount of /alt_inst/var/adm/ras/livedump
forced unmount of /alt_inst/var
forced unmount of /alt_inst/var
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/home
forced unmount of /alt_inst/home
forced unmount of /alt_inst/admin
forced unmount of /alt_inst/admin
forced unmount of /alt_inst
forced unmount of /alt_inst
Fixing file system superblocks...
Bootlist is set to the boot disk: hdisk1 blv=hd5
[root@sls-demo-test2 data]#
```

11. *After completion of the restore check the bootlist to make sure when we reboot the OS it will boot from the new volume.*

```
[root@sls-demo-test2 data]#bootlist -m normal -o
hdisk1 blv=hd5 pathid=0
hdisk1 blv=hd5 pathid=1
hdisk1 blv=hd5 pathid=2
hdisk1 blv=hd5 pathid=3
```

12. *Reboot the VM. After the reboot, login into the new VM and check the disk of the new VM.*

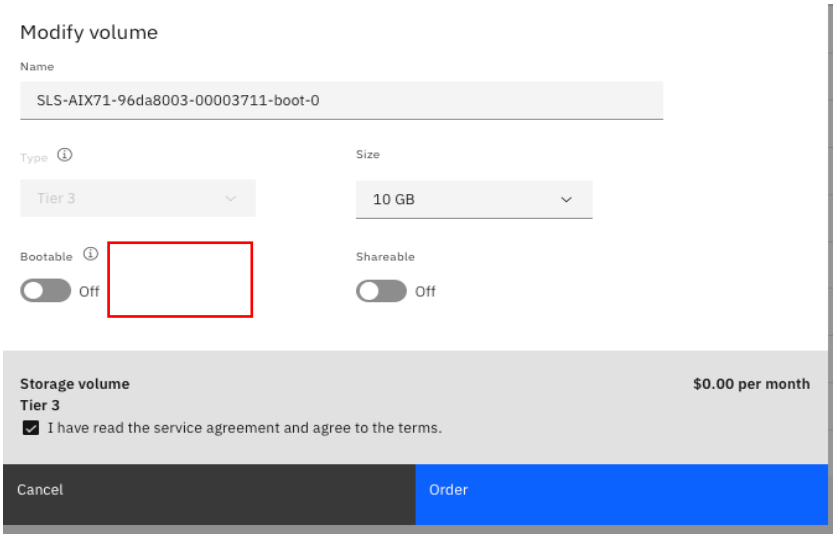
Note: Reboot will take several minutes due to device reconfiguration

```
Berjis-MBP15:~ berjis$ ssh root@161.156.153.188
Last unsuccessful login: Tue Dec 10 13:42:53 CST 2019 on ssh from infsvrlnx.ibmpowercloud.com
Last login: Wed Apr 15 15:05:02 CDT 2020 on /dev/pts/0 from 47.185.145.228
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 7.1!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
[root@sls-demo-test2 /]#lspv
hdisk0          00f6db0a6c7aece5                    old_rootvg
hdisk1          00c63e307f557a96                    rootvg          active
[root@sls-demo-test2 /]#
```

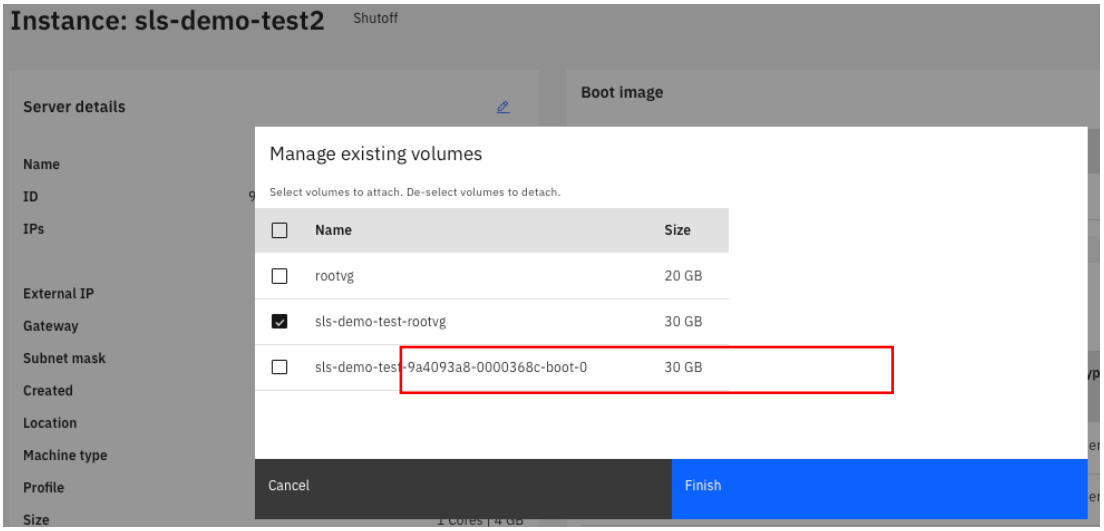
13. *You can then detach the original or helper disk by running exportvg and rmdev as shown below.*

```
[root@sls-demo-test2 /]#exportvg old_rootvg
[root@sls-demo-test2 /]#lspv
hdisk0          00f6db0a6c7aece5                    None
hdisk1          00c63e307f557a96                    rootvg          active
[root@sls-demo-test2 /]#rmdev -dl hdisk0 -R
hdisk0 deleted
[root@sls-demo-test2 /]#lspv
hdisk1          00c63e307f557a96                    rootvg          active
[root@sls-demo-test2 /]#
```

14. *We will now remove the boot disk flag on the disk we will be detaching by going to Storage Volumes and editing the disk by clicking the pencil icon.*



15. *You can now detach the disk, by selecting the virtual server instances and click on the instance, then click on "Manage existing volumes" and de-select the disk you want to detach.*



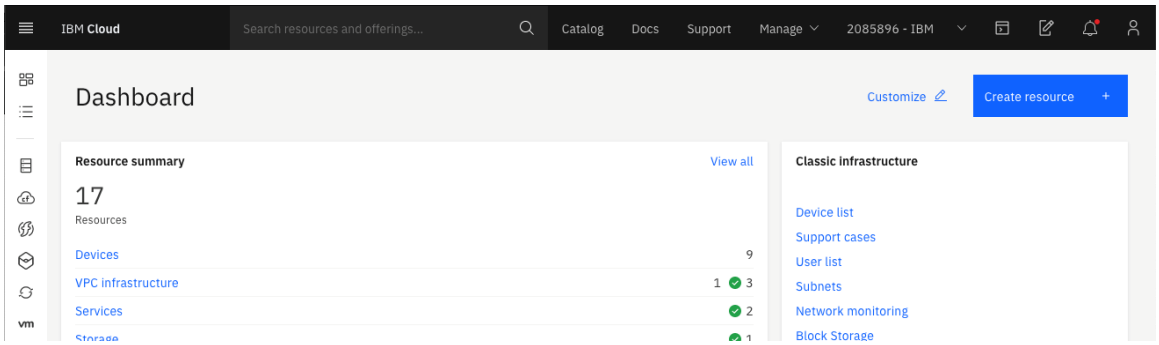
Note: Make sure the boot disk flag is turned off for the disk you are detaching

Transfer System Backup Using Cloud Object Storage

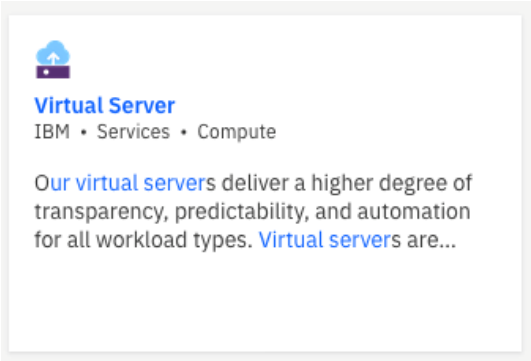
For this approach, you will transfer your mksysb image to Cloud Object Storage and the use a Linux Virtual Server Instance (VSI) in IBM Cloud to transfer the image and make it available to restore into Power VS.

Create a Linux VSI for Staging

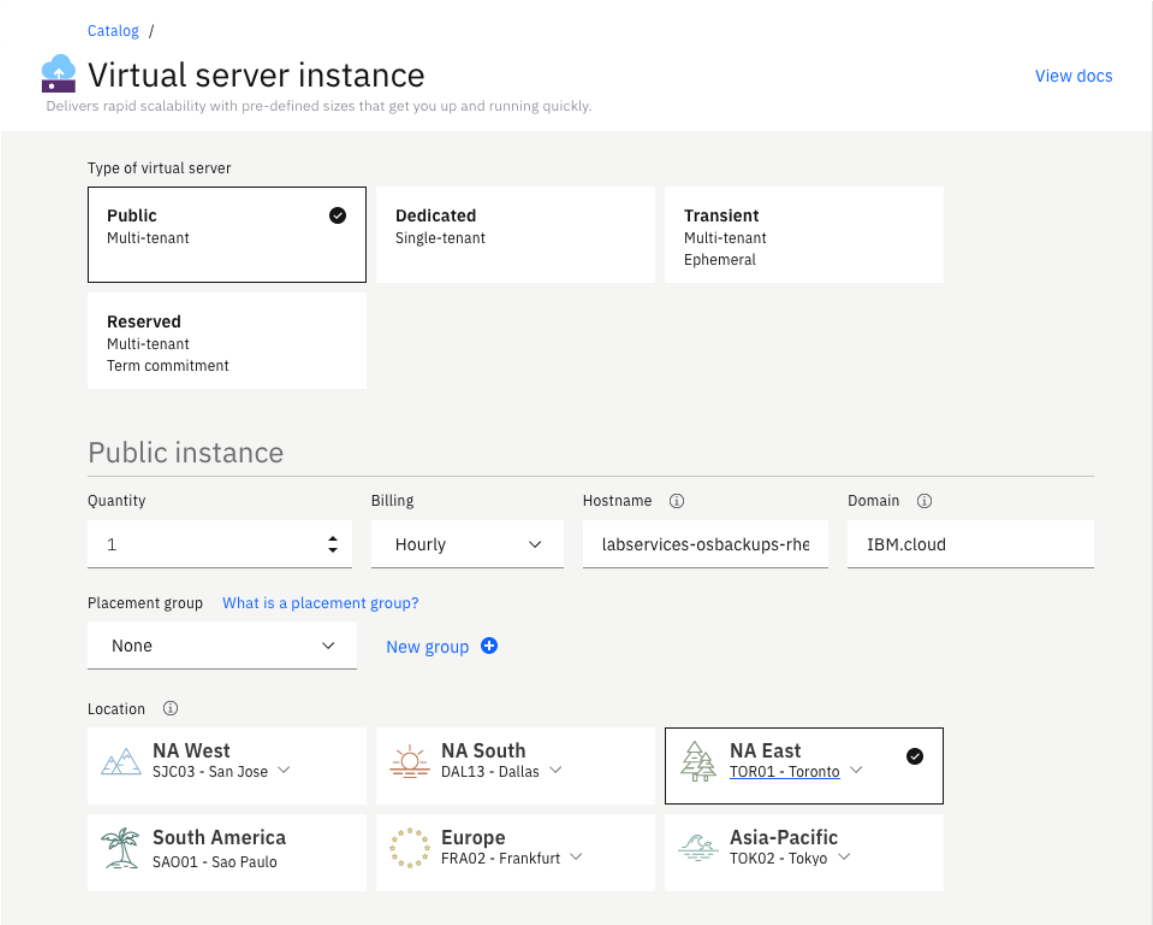
To begin, create your Linux VSI with appropriate resources. From the IBM Cloud dashboard, click the blue Create resource button in the upper right corner.



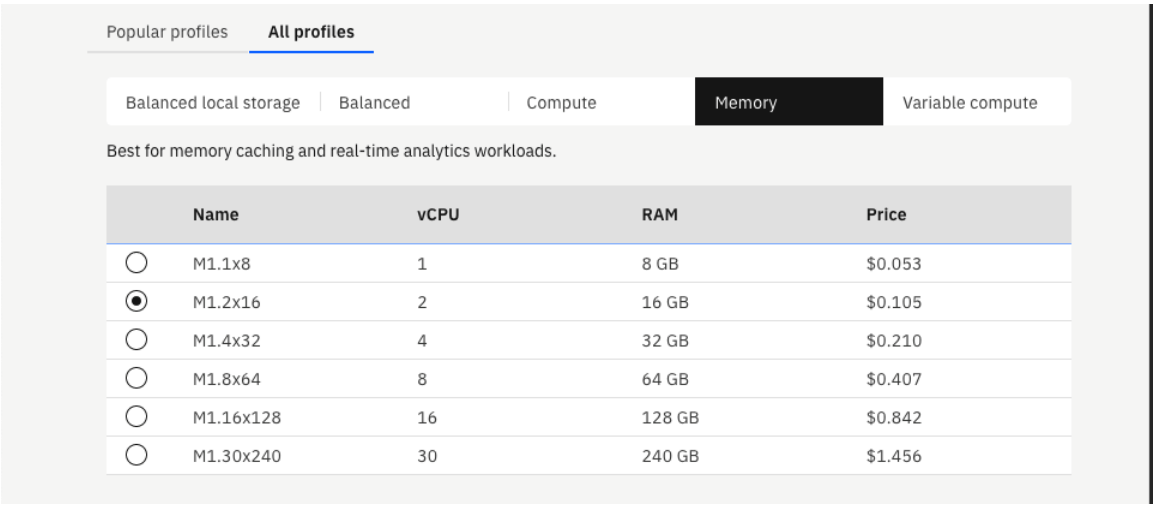
Search for and select Virtual Server.



You can select the option for Public. Give your VSI a meaningful Hostname and select the same region as your Power VS environment. Then scroll down to select further options.

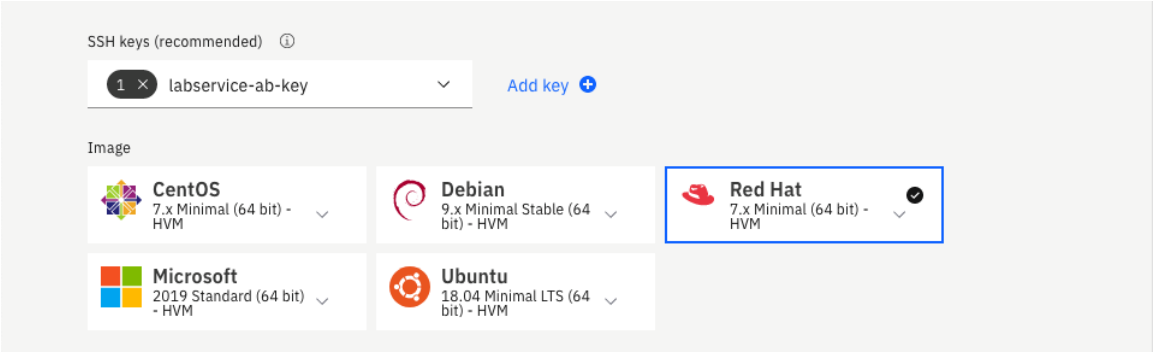


Click on All profiles and then click on the option for Memory and select the M1.2x16 profile. Depending on your actual usage, you may determine you need additional resources, but this is a good starting point.

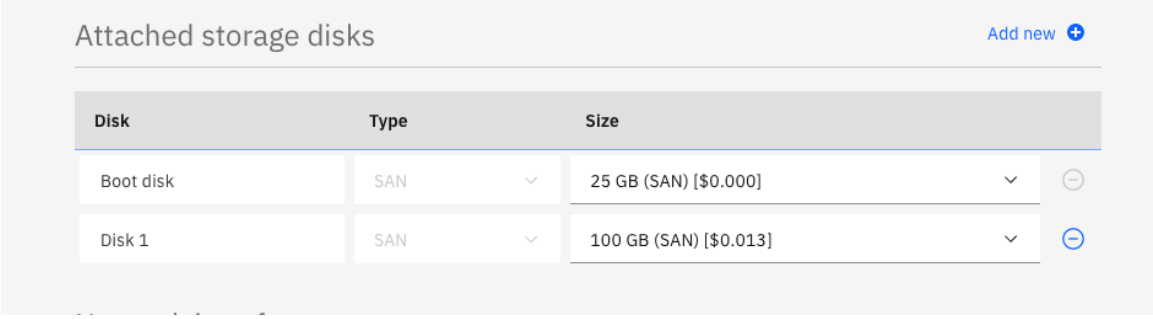


	Name	vCPU	RAM	Price
○	M1.1x8	1	8 GB	$0.053
◉	M1.2x16	2	16 GB	$0.105
○	M1.4x32	4	32 GB	$0.210
○	M1.8x64	8	64 GB	$0.407
○	M1.16x128	16	128 GB	$0.842
○	M1.30x240	30	240 GB	$1.456

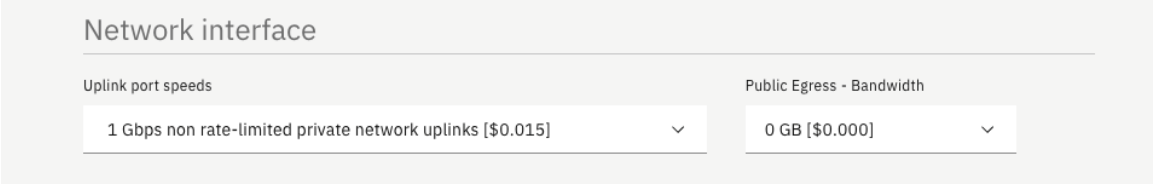
Below that, select an ssh key, if you have one configured, for more convenient console access and choose a Red Hat operating system.



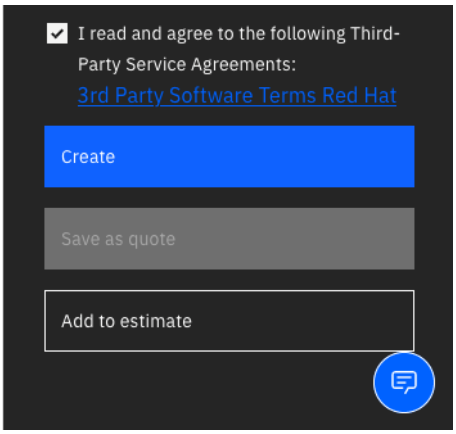
You'll need to add an additional disk to allow space to stage mksysb images. Click the Add New button to the right of Attached Storage Disks and create an addition 100 GB disk.



Finally, change the network Uplink port speeds to 1 Gbps private.

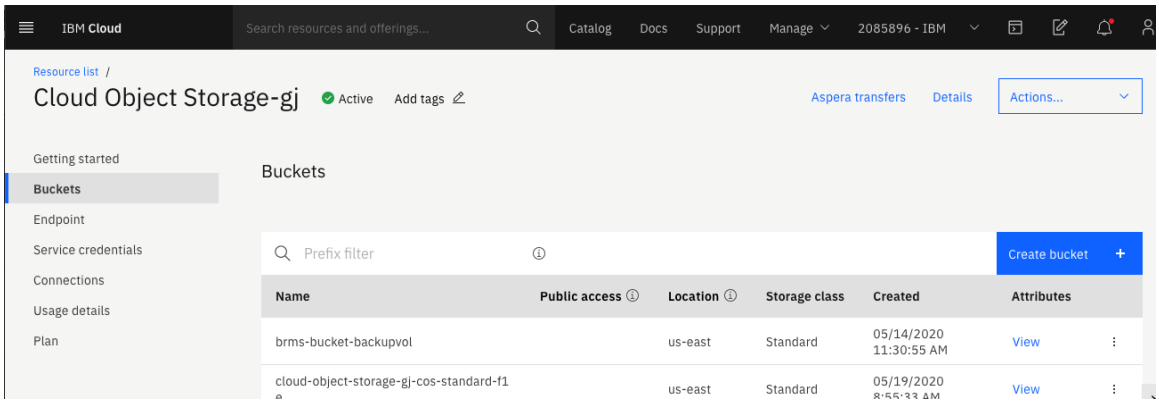


Then, in the right column, select the check box to accept the Red Hat service agreement and click the Create button. Wait a few minutes as your new VSI is provisioned.

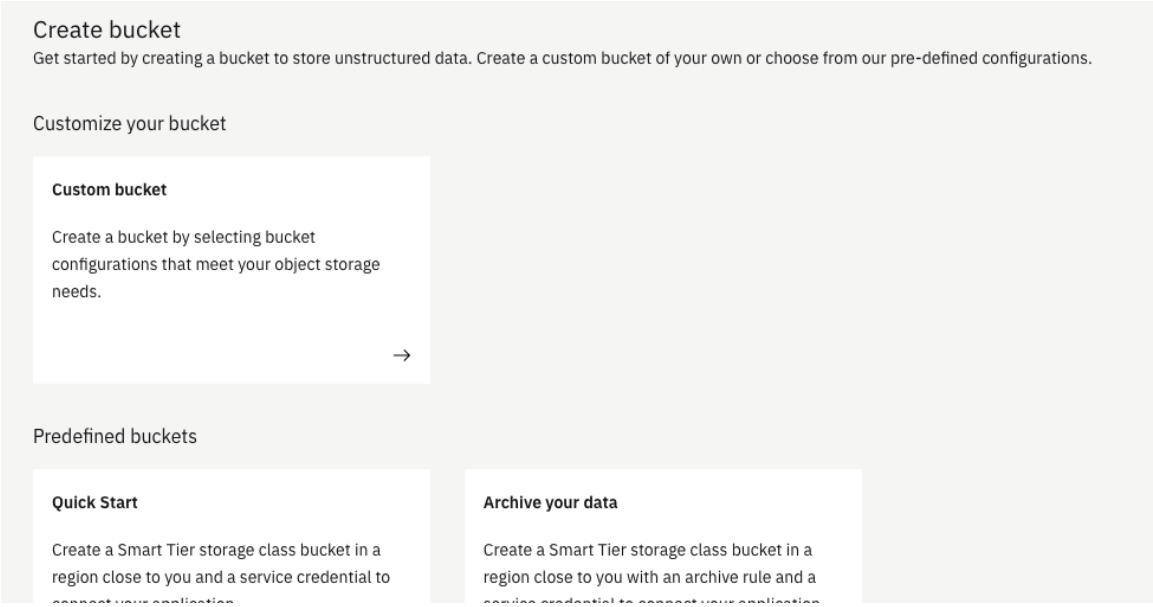


Create a Cloud Object Storage Bucket

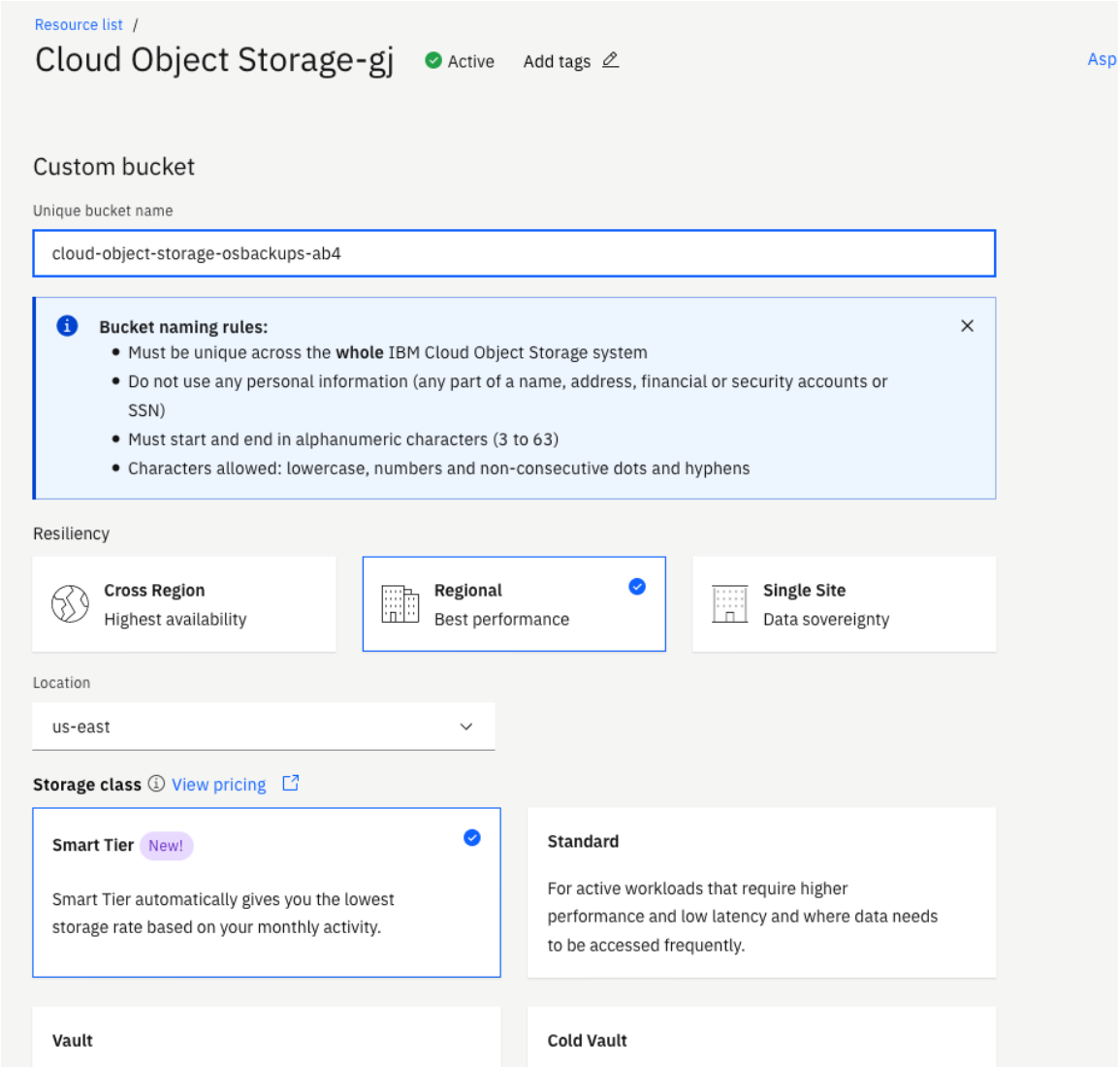
You'll need a Cloud Object Storage bucket to store your mksysb files. Navigate to the Cloud Object Storage resource in the IBM Cloud GUI. Click the blue Create Bucket button on the right side of the Buckets screen.



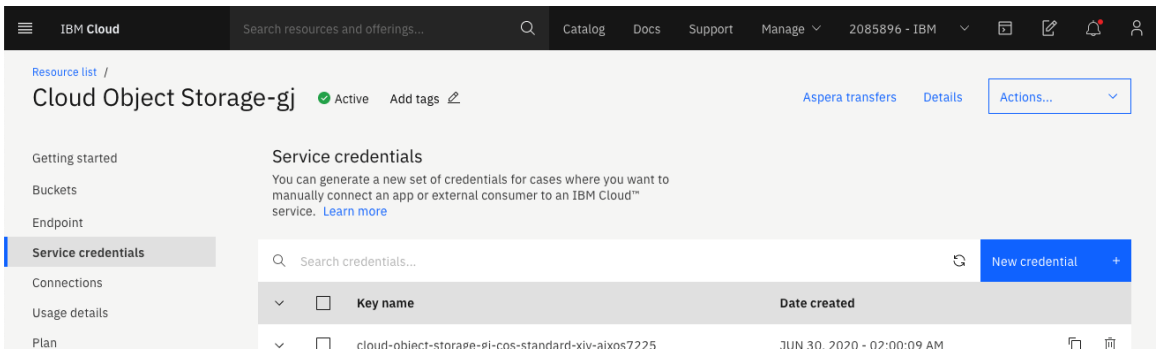
Select the option for Custom Bucket.



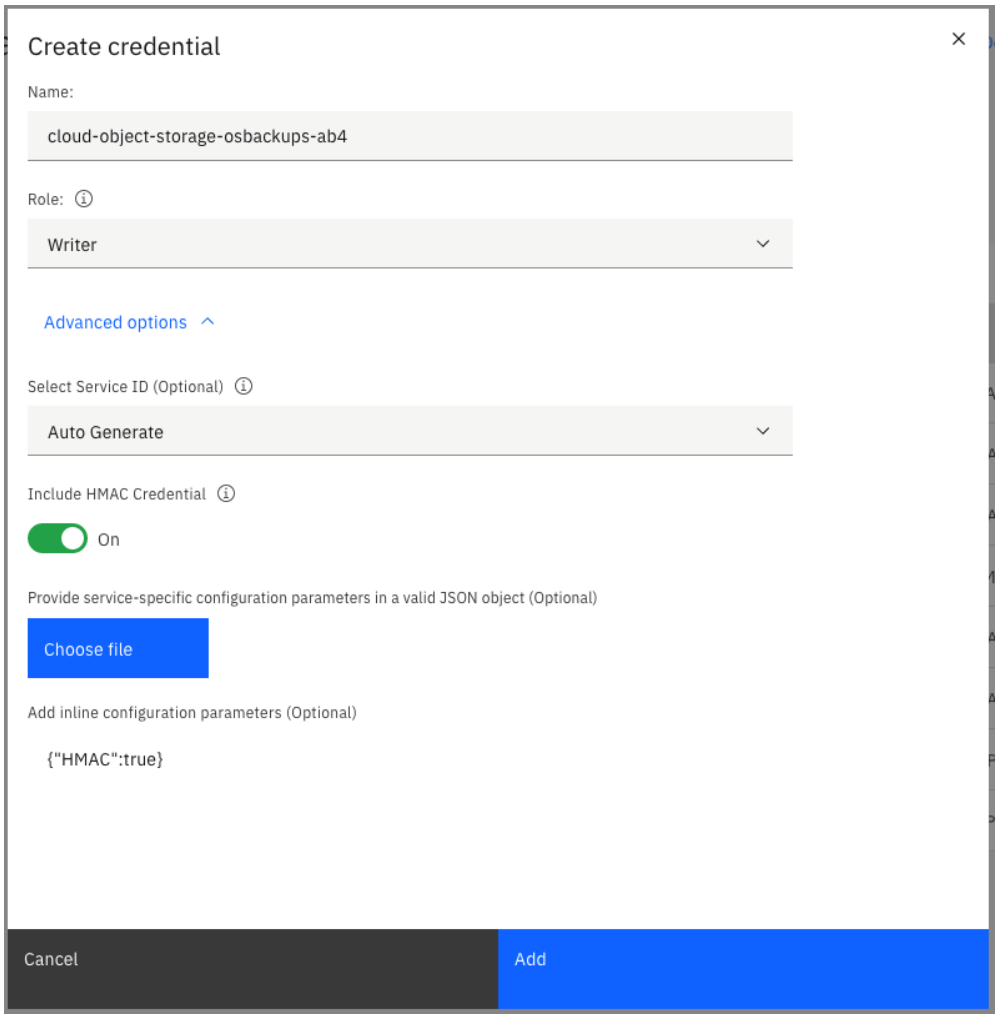
Give your bucket a meaningful name and ensure your Resiliency, Location and Storage Class options are appropriate. Then scroll to the bottom and click the Create Bucket button.



Now click the Service Credentials item in the left column to create a key to access the bucket.



Give your Service Credential a meaningful name. Then, click Advanced Options and select the option to Include HMAC credential. Then click Add.



Configure s3fs-fuse in the Linux VSI

After you've completed that configuration, log in to the Linux VSI you've created. Next you will install s3fs-fuse, which will allow you to attach your Cloud Object Storage bucket as a filesystem. First use yum to update packages in the VSI to current levels.

```
[root@labservices-osbackups-rhel-ab4 ~]# yum -y update
Loaded plugins: product-id, search-disabled-repos, subscription-manager
rhel-7-server-optional-rpms                                      | 2.0 kB  00:00:00
rhel-7-server-rpms                                               | 2.0 kB  00:00:00
rhel-7-server-supplementary-rpms                                 | 2.0 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager.x86_64 1:1.18.4-3.el7 will be an update
---> Package NetworkManager-config-server.noarch 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-config-server.noarch 1:1.18.4-3.el7 will be an update
---> Package NetworkManager-libnm.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-libnm.x86_64 1:1.18.4-3.el7 will be an update
---> Package NetworkManager-ppp.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-ppp.x86_64 1:1.18.4-3.el7 will be an update
---> Package NetworkManager-team.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-team.x86_64 1:1.18.4-3.el7 will be an update
---> Package NetworkManager-tui.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-tui.x86_64 1:1.18.4-3.el7 will be an update
---> Package acl.x86_64 0:2.2.51-14.el7 will be updated
---> Package acl.x86_64 0:2.2.51-15.el7 will be an update
---> Package avahi-autoipd.x86_64 0:0.6.31-19.el7 will be updated
---> Package avahi-autoipd.x86_64 0:0.6.31-20.el7 will be an update
---> Package avahi-libs.x86_64 0:0.6.31-19.el7 will be updated
---> Package avahi-libs.x86_64 0:0.6.31-20.el7 will be an update

... Many lines skipped ...

  sudo.x86_64 0:1.8.23-9.el7
  systemd.x86_64 0:219-73.el7_8.8
  systemd-libs.i686 0:219-73.el7_8.8
  systemd-libs.x86_64 0:219-73.el7_8.8
  systemd-sysv.x86_64 0:219-73.el7_8.8
  teamd.x86_64 0:1.29-1.el7
  tuned.noarch 0:2.11.0-8.el7
  tzdata.noarch 0:2020a-1.el7
  util-linux.x86_64 0:2.23.2-63.el7
  yum.noarch 0:3.4.3-167.el7

Complete!
[root@labservices-osbackups-rhel-ab4 ~]#
```

Then use yum to install the necessary tools to build the s3fs-fuse package.

```
[root@labservices-osbackups-rhel-ab4 ~]# yum -y install automake fuse fuse-devel gcc-
c++ git libcurl-devel libxml2-devel make openssl-devel unzip
Loaded plugins: product-id, search-disabled-repos, subscription-manager
rhel-7-server-optional-rpms                                      | 2.0 kB  00:00:00
rhel-7-server-rpms                                               | 2.0 kB  00:00:00
rhel-7-server-supplementary-rpms                                 | 2.0 kB  00:00:00
Package 1:make-3.82-24.el7.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package automake.noarch 0:1.13.4-3.el7 will be installed
--> Processing Dependency: perl >= 5.006 for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: autoconf >= 2.65 for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(warnings) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(vars) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(threads) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(strict) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(constant) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(Thread::Queue) for package: automake-1.13.4-
3.el7.noarch
--> Processing Dependency: perl(TAP::Parser) for package: automake-1.13.4-3.el7.noarch
--> Processing Dependency: perl(POSIX) for package: automake-1.13.4-3.el7.noarch

... Many lines skipped ...

  perl-macros.x86_64 4:5.16.3-295.el7
  perl-parent.noarch 1:0.225-244.el7
  perl-podlators.noarch 0:2.5.1-3.el7
  perl-threads.x86_64 0:1.87-4.el7
  perl-threads-shared.x86_64 0:1.43-6.el7
  rsync.x86_64 0:3.1.2-10.el7
  xz-devel.x86_64 0:5.2.2-1.el7
  zlib-devel.x86_64 0:1.2.7-18.el7

Complete!
[root@labservices-osbackups-rhel-ab4 ~]#
```

Now, visit https://github.com/s3fs-fuse/s3fs-fuse and download the code for s3fs-fuse as a zip. Transfer that zip to your VSI and unpack it.

```
[root@labservices-osbackups-rhel-ab8 s3fs]# unzip s3fs-fuse-master.zip
Archive:  s3fs-fuse-master.zip
e0a38adaf6cec3f413bfe0bc45869bcf33301f19
   creating: s3fs-fuse-master/
  inflating: s3fs-fuse-master/.clang-tidy
  inflating: s3fs-fuse-master/.gitattributes

... Several lines skipped ...

  inflating: s3fs-fuse-master/test/sample_ahbe.conf
  inflating: s3fs-fuse-master/test/sample_delcache.sh
  inflating: s3fs-fuse-master/test/small-integration-test.sh
  inflating: s3fs-fuse-master/test/test-utils.sh
  inflating: s3fs-fuse-master/test/ut_test.py
  inflating: s3fs-fuse-master/test/write_multiple_offsets.py
[root@labservices-osbackups-rhel-ab8 s3fs]# ls
s3fs-fuse-master  s3fs-fuse-master.zip
[root@labservices-osbackups-rhel-ab8 s3fs]#
```

Next change to the s3fs-fuse-master directory and autogen.sh, configure, make and make install to build and install s3fs-fuse.

```
[root@labservices-osbackups-rhel-ab4 ~]# cd s3fs-fuse-master/
[root@labservices-osbackups-rhel-ab4 s3fs-fuse]# ./autogen.sh
--- Make commit hash file -------
--- Finished commit hash file ---
--- Start autotools -------------
configure.ac:26: installing './config.guess'
configure.ac:26: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
src/Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
--- Finished autotools ----------
[root@labservices-osbackups-rhel-ab4 s3fs-fuse]# ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes

... Many lines skipped ...

checking github short commit hash... 62c8be8
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating test/Makefile
config.status: creating doc/Makefile
config.status: creating config.h
config.status: executing depfiles commands
[root@labservices-osbackups-rhel-ab4 s3fs-fuse]# make
make  all-recursive
make[1]: Entering directory `/root/s3fs-fuse'
Making all in src
make[2]: Entering directory `/root/s3fs-fuse/src'
g++ -DHAVE_CONFIG_H -I. -I..  -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -
I/usr/include/libxml2      -g -O2 -Wall -D_FILE_OFFSET_BITS=64 -D_FORTIFY_SOURCE=2 -MT

... Many lines skipped ...

make[2]: Entering directory `/root/s3fs-fuse'
make[2]: Leaving directory `/root/s3fs-fuse'
make[1]: Leaving directory `/root/s3fs-fuse'
[root@labservices-osbackups-rhel-ab4 s3fs-fuse]# make install
Making install in src
make[1]: Entering directory `/root/s3fs-fuse/src'
make[2]: Entering directory `/root/s3fs-fuse/src'
 /usr/bin/mkdir -p '/usr/local/bin'

... Several lines skipped ...

make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/root/s3fs-fuse'
make[1]: Leaving directory `/root/s3fs-fuse'
[root@labservices-osbackups-rhel-ab4 s3fs-fuse]#
```

Now you need to configure access to your bucket. In the Cloud Object Storage section of the IBM Cloud console, locate the Service Credential you created earlier.

cloud-object-storage-osbackups-ab4 JUL 11, 2020 - 10:55:28 AM


```
{
  "apikey": "6ee_458GISoUqC3KlxKO1Swvx6ncu4xUi0W1XCK7aaHD",
  "cos_hmac_keys": {
    "access_key_id": "9eec0f82e7dd4a72a6fa19f0bd67d657",
    "secret_access_key": "c22701d0d39517b860699ba0d8cab1ec23a7a04d71ed1d05"
  },
  "endpoints": "https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints",
  "iam_apikey_description": "Auto-generated for key 9eec0f82-e7dd-4a72-a6fa-19f0bd67d657",
  "iam_apikey_name": "cloud-object-storage-osbackups-ab4",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/06d2a1ecba244622a0fb88efb4843fb4::serviceid:
ServiceId-42e92280-255e-420d-a7d9-4e1d2d98a3d6",
  "resource_instance_id": "crn:v1:bluemix:public:cloud-object-storage:global:a/06d2a1ecba244622a0fb88efb48
43fb4:3513c7a1-690e-4fdf-9ec5-fa679037e8db::"
}
```


Create a file /etc/passwd-s3fs containing your access_key_id and secret_access_key separated by a colon.

```
9eec0f82e7dd4a72a6fa19f0bd67d657:c22701d0d39517b860699ba0d8cab1ec23a7a04d71ed1d05
~
~
```

Set the permissions on that file.

```
[root@labservices-osbackups-rhel-ab4 ~]# chmod 600 /etc/passwd-s3fs
[root@labservices-osbackups-rhel-ab4 ~]#
```

Now create a mount point to attach your bucket and use the s3fs command to attach the storage. You'll need the name of the bucket and the url of the private Cloud Object Storage endpoint for the appropriate region. Use df to confirm the mount succeeded.

```
[root@labservices-osbackups-rhel-ab4 ~]# mkdir /cosbucket
[root@labservices-osbackups-rhel-ab4 ~]# s3fs cloud-object-storage-osbackups-ab4
/cosbucket -o passwd_file=/etc/passwd-s3fs -o url=https://s3.private.us-east.cloud-
object-storage.appdomain.cloud -o use_path_request_style -o dbglevel=info -o
allow_other
[root@labservices-osbackups-rhel-ab4 ~]# df -h | grep s3fs
s3fs            256T     0  256T   0% /cosbucket
[root@labservices-osbackups-rhel-ab4 ~]#
```

Create and Export Staging File System

Next, you'll need to format and mount your staging disk. You can use fdisk to determine the name of the intended disk. Look for the disk that is around 100 GB, in this case /dev/xvcd.

```
[root@labservices-osbackups-rhel-ab8 ~]# fdisk -l

Disk /dev/xvda: 26.8 GB, 26843545600 bytes, 52428800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0000cece

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *        2048     2099199     1048576   83  Linux
/dev/xvda2         2099200    52428799    25164800   83  Linux

Disk /dev/xvdc: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/xvdb: 2147 MB, 2147483648 bytes, 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00025cdb

    Device Boot      Start         End      Blocks   Id  System
/dev/xvdb1              63     4192964     2096451   82  Linux swap / Solaris
[root@labservices-osbackups-rhel-ab8 ~]#
```

Format that disk using mkfs.xfs.

```
[root@labservices-osbackups-rhel-ab8 ~]# mkfs.xfs /dev/xvdc
meta-data=/dev/xvdc              isize=512    agcount=4, agsize=6553600 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=26214400, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=12800, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@labservices-osbackups-rhel-ab8 ~]#
```

Create a mount point to attach the new disk.

```
[root@labservices-osbackups-rhel-ab8 ~]# mkdir /stage
```

Then edit /etc/fstab and add a line to mount that disk on the mount point.

```
#
# /etc/fstab
# Created by anaconda on Thu Oct  3 14:41:18 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=b894c135-27a1-4f7c-8cb1-8b3a69a05491 /                       ext3
defaults,noatime        1 1
UUID=1205ee90-24ba-4bed-af8d-7f9bf36008ed /boot                   ext3
defaults,noatime        1 2
LABEL=SWAP-xvdb1 swap swap    defaults        0 0

# Filesystem for mksysb staging
/dev/xvdc    /tsm     xfs    defaults    1 2
```

Use mount -a to mount the new file system and df -h to check that it is available.

```
[root@labservices-osbackups-rhel-ab8 ~]# mount -a
[root@labservices-osbackups-rhel-ab8 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G     0  7.8G   0% /dev
tmpfs           7.8G     0  7.8G   0% /dev/shm
tmpfs           7.8G   17M  7.8G   1% /run
tmpfs           7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/xvda2       24G  3.5G   19G  16% /
/dev/xvda1      976M  155M  770M  17% /boot
tmpfs           1.6G     0  1.6G   0% /run/user/0
s3fs            256T     0  256T   0% /cosbucket
/dev/xvdc       100G   33M  100G   1% /stage
[root@labservices-osbackups-rhel-ab8 ~]#
```

Next, you can configure nfs to share your staging file system with your AIX VSIs. Begin by using yum to install the nfs utilities.

```
[root@labservices-osbackups-rhel-ab4 ~]# yum install nfs-utils
Loaded plugins: product-id, search-disabled-repos, subscription-manager
rhel-7-server-optional-rpms                                       | 2.0 kB  00:00:00
rhel-7-server-rpms                                                | 2.0 kB  00:00:00
rhel-7-server-supplementary-rpms                                  | 2.0 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package nfs-utils.x86_64 1:1.3.0-0.66.el7 will be installed
--> Processing Dependency: libtirpc >= 0.2.4-0.7 for package: 1:nfs-utils-1.3.0-
0.66.el7.x86_64
--> Processing Dependency: gssproxy >= 0.7.0-3 for package: 1:nfs-utils-1.3.0-
0.66.el7.x86_64

... Many lines skipped ...

Installed:
  nfs-utils.x86_64 1:1.3.0-0.66.el7

Dependency Installed:
  gssproxy.x86_64 0:0.7.0-28.el7              keyutils.x86_64 0:1.5.8-3.el7
  libbasicobjects.x86_64 0:0.1.1-32.el7       libcollection.x86_64 0:0.7.0-32.el7
  libevent.x86_64 0:2.0.21-4.el7              libini_config.x86_64 0:1.3.1-32.el7
  libnfsidmap.x86_64 0:0.25-19.el7            libpath_utils.x86_64 0:0.2.1-32.el7
  libref_array.x86_64 0:0.1.5-32.el7          libtirpc.x86_64 0:0.2.4-0.16.el7
  libverto-libevent.x86_64 0:0.2.5-4.el7      quota.x86_64 1:4.01-19.el7
  quota-nls.noarch 1:4.01-19.el7              rpcbind.x86_64 0:0.2.0-49.el7
  tcp_wrappers.x86_64 0:7.6-77.el7

Complete!
[root@labservices-osbackups-rhel-ab4 ~]#
```

Then edit /etc/exports to share the filesystem where your Cloud Object Storage bucket is mounted. Ensure you specify the correct subnet information for your PowerVS environment.

```
/stage 192.168.50.0/24(rw,no_root_squash,insecure)
~
~
```

Now start the nfs-server service.

```
[root@labservices-osbackups-rhel-ab4 ~]# systemctl start nfs-server
[root@labservices-osbackups-rhel-ab4 ~]#
```

Back Up Your On-Premise AIX System and Upload the Image

Take a mksysb backup of your AIX VSI. Enter smit mksysb to begin. Enter a meaningful filename in the mount you made earlier in the Backup Device or File field. Ensure Expand /tmp if needed is set to yes. Then press Enter to begin the backup.

```
                 Back Up This System to Tape/File or UDFS capable media

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
    WARNING:  Execution of the mksysb command will
              result in the loss of all material
              previously stored on the selected
              output medium. This command backs
              up only rootvg volume group.

* Backup DEVICE or FILE                              [/stage/aix72backup1.mksysb] +/
  Create MAP files?                                   no                           +
  Create backup using snapshots?                      no                           +
  EXCLUDE files?                                      no                           +
  Exclude WPAR file systems?                          no                           +
  Location of File System Exclusion List             []                             /
  List files as they are backed up?                   no                           +
  Verify readability if tape device?                  no                           +
  Generate new /image.data file?                      yes                          +
  EXPAND /tmp if needed?                              yes                          +
  Disable software packing of backup?                 no                           +
  Backup extended attributes?                         yes                          +
  Number of BLOCKS to write in a single output       []                             #
     (Leave blank to use a system default)
  Location of existing mksysb image                  []                             /
  File system to use for temporary work space        []                             /
     (If blank, /tmp will be used.)
  Backup encrypted files?                             yes                          +
  Back up DMAPI filesystem files?                     yes                          +
  Build new alt_disk_install boot_image?              no                           +

F1=Help              F2=Refresh            F3=Cancel            F4=List
F5=Reset             F6=Command            F7=Edit              F8=Image
F9=Shell             F10=Exit              Enter=Do
```

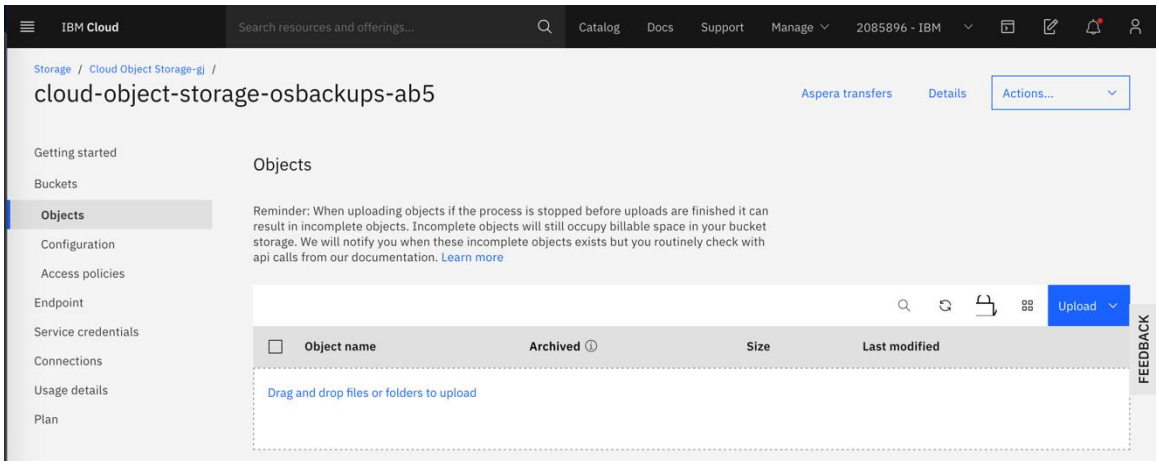
Once your backup completes, you can exit smit.

```
                                    COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.


Creating information file (/image.data) for rootvg.

Creating list of files to back up

Backing up 56586 files.............................
4542 of 56586 files backed up (8%)..............................
5139 of 56586 files backed up (9%)..............................
5861 of 56586 files backed up (10%)...........................

56586 of 56586 files backed up (100%)
0512-038 mksysb: Backup Completed Successfully.

F1=Help              F2=Refresh            F3=Cancel            F6=Command
F8=Image             F9=Shell              F10=Exit             /=Find
n=Find Next
```

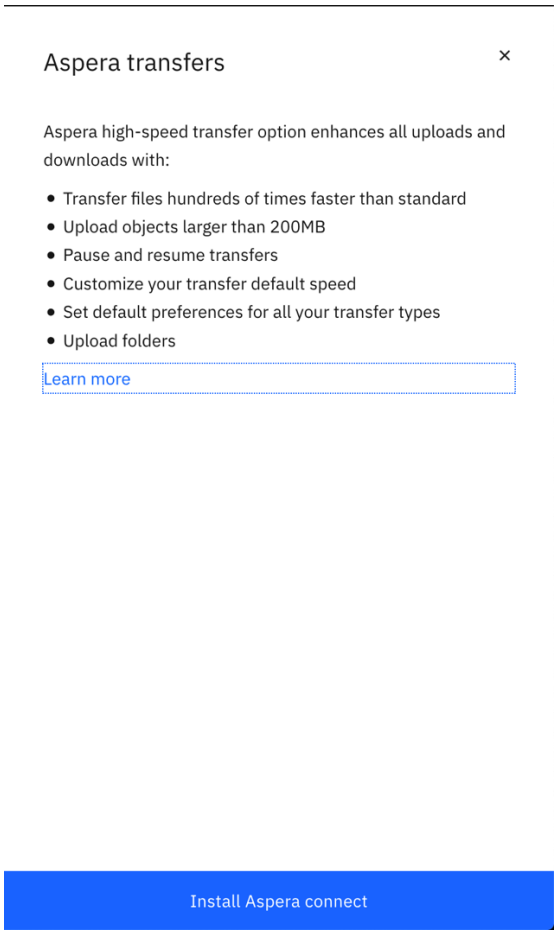
Now copy the backup to your workstation system. You could use some shared network storage to facilitate this, but for this example scp is used.

```
aaron@Aarons-Work-MacBook-Air Desktop % scp
root@ablnx1.rchland.ibm.com:/stage/aix72backup1.mksysb .
root@ablnx1.rchland.ibm.com's password:
aix72backup1.mksysb                                 100% 1482MB   1.4MB/s   17:34
aaron@Aarons-Work-MacBook-Air Desktop %
```

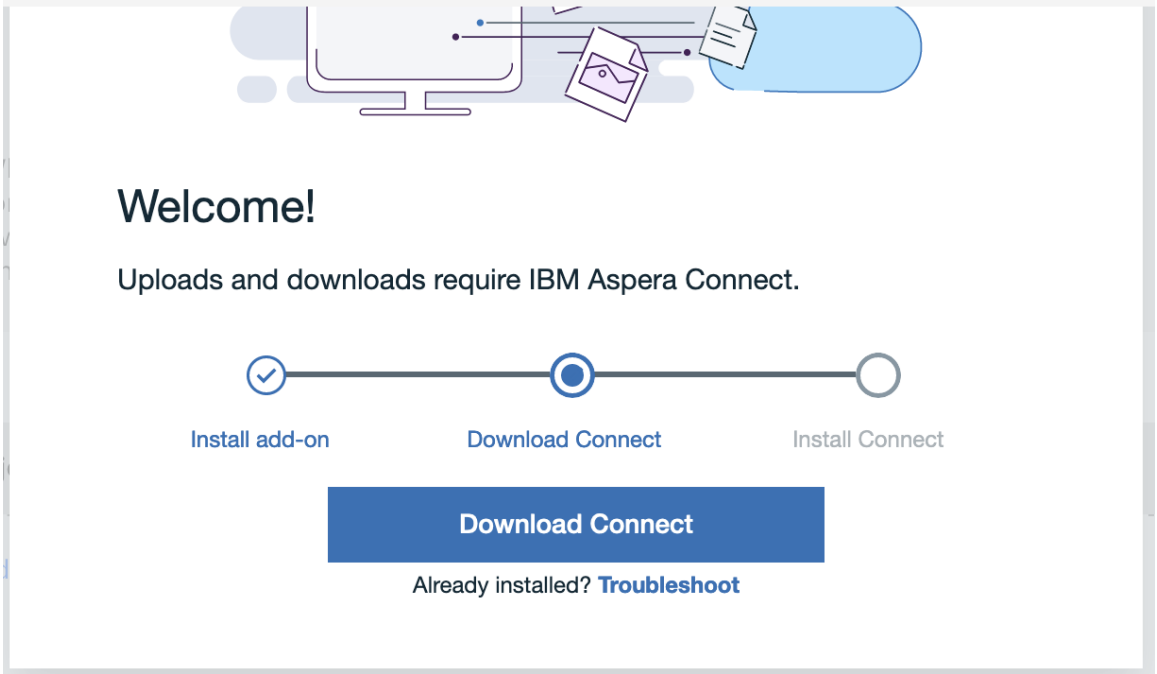
Now, navigate to your bucket in the Cloud Object Storage GUI. Click the Aspera transfers link near the top right.



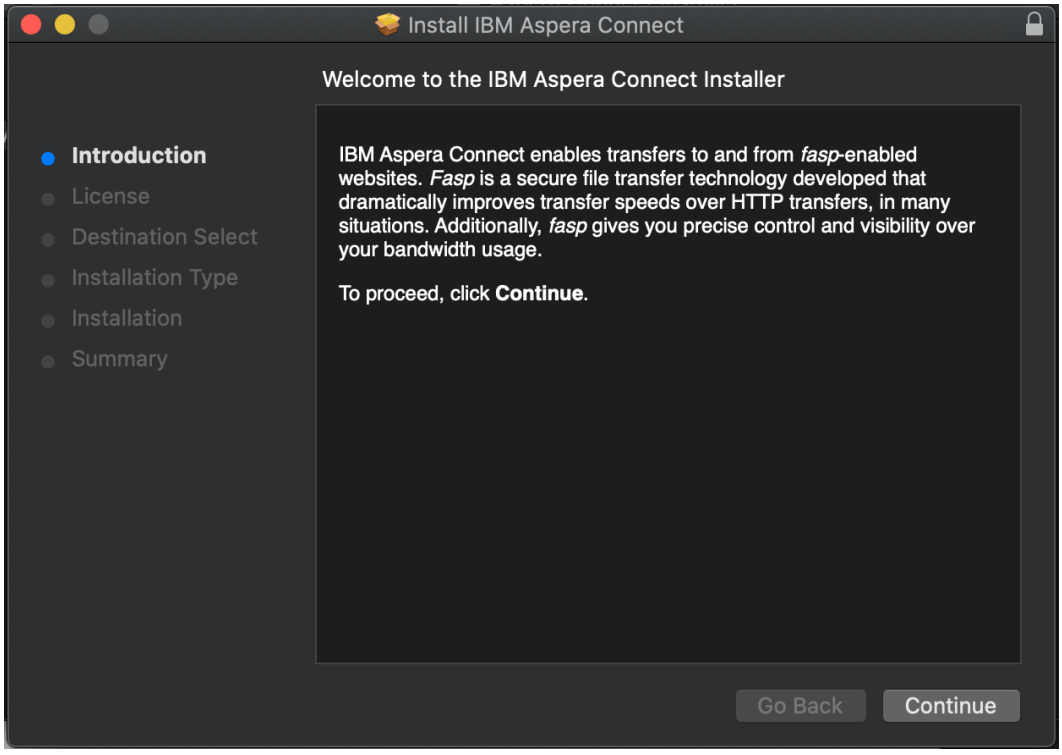
Click the option to Install Aspera Connect at the bottom of the pane that appears on the right.



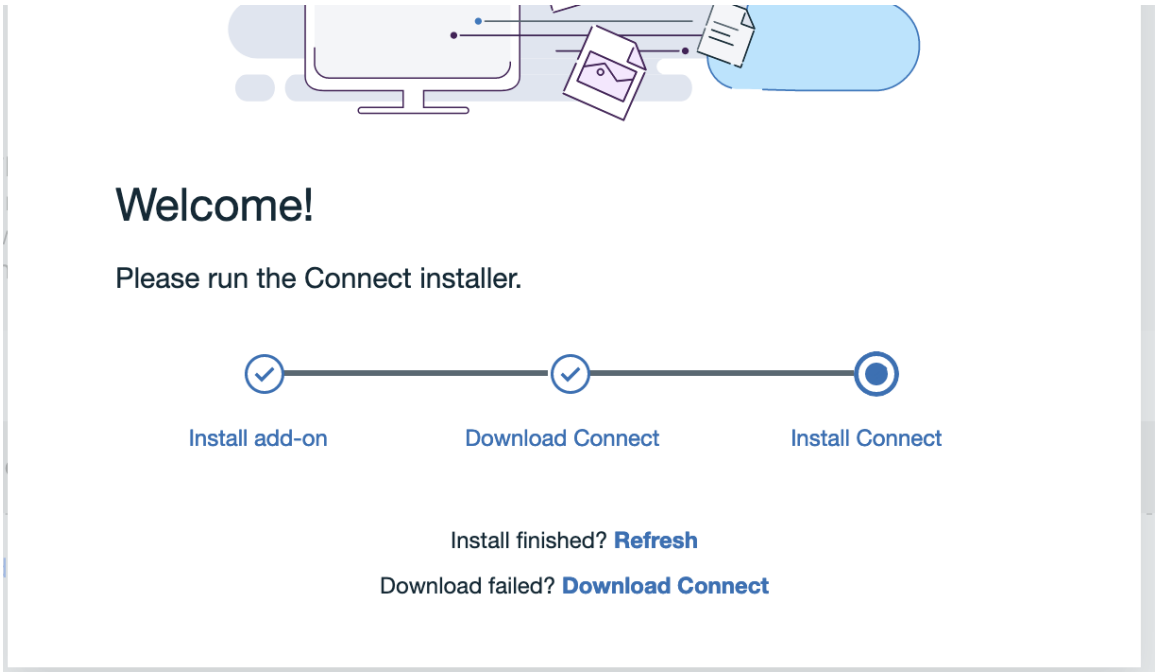
Then click the Download Connect button that appears in the center.



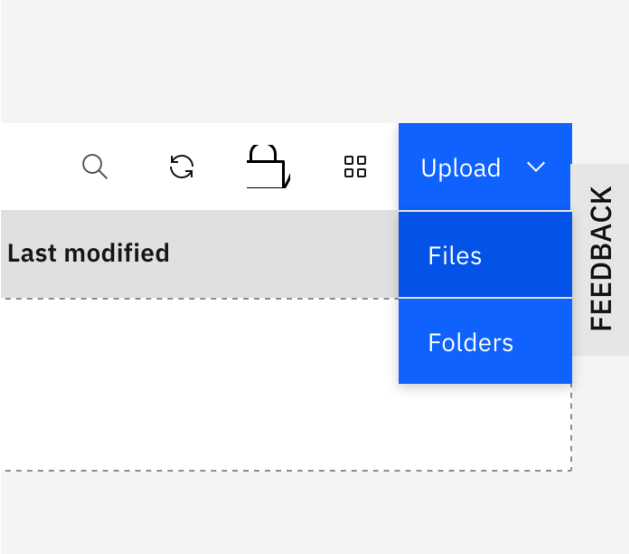
Use the appropriate method for your operating system to launch the installer and complete the required installation steps.



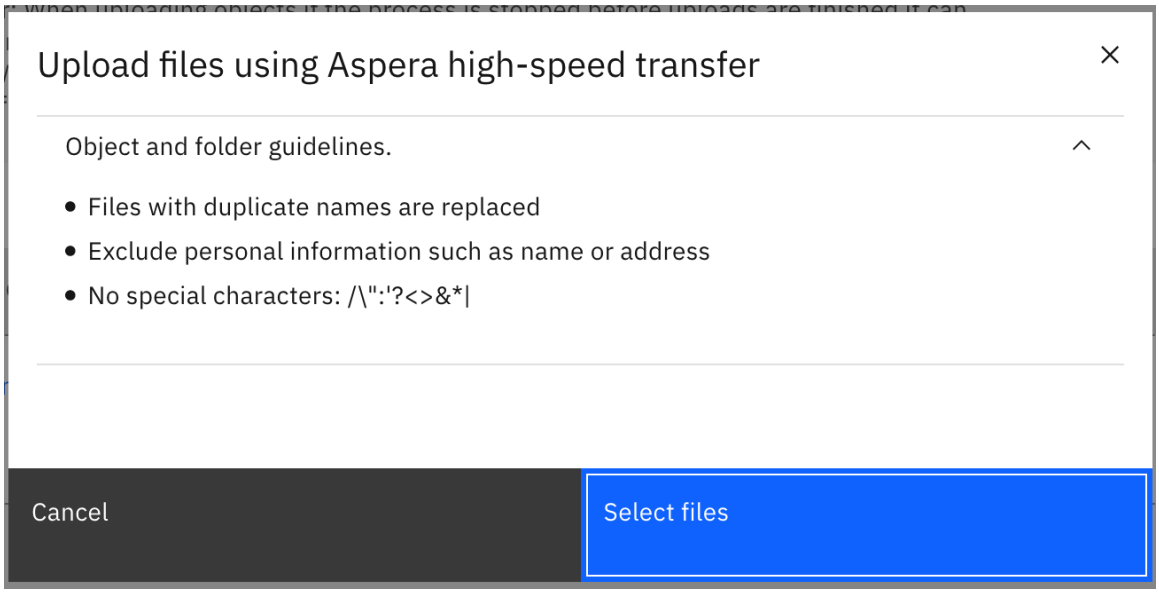
Once the installation is complete, return to the browser and click the link to refresh in the center of the page. The page will reload and the Aspera Connect client will launch automatically in the background.



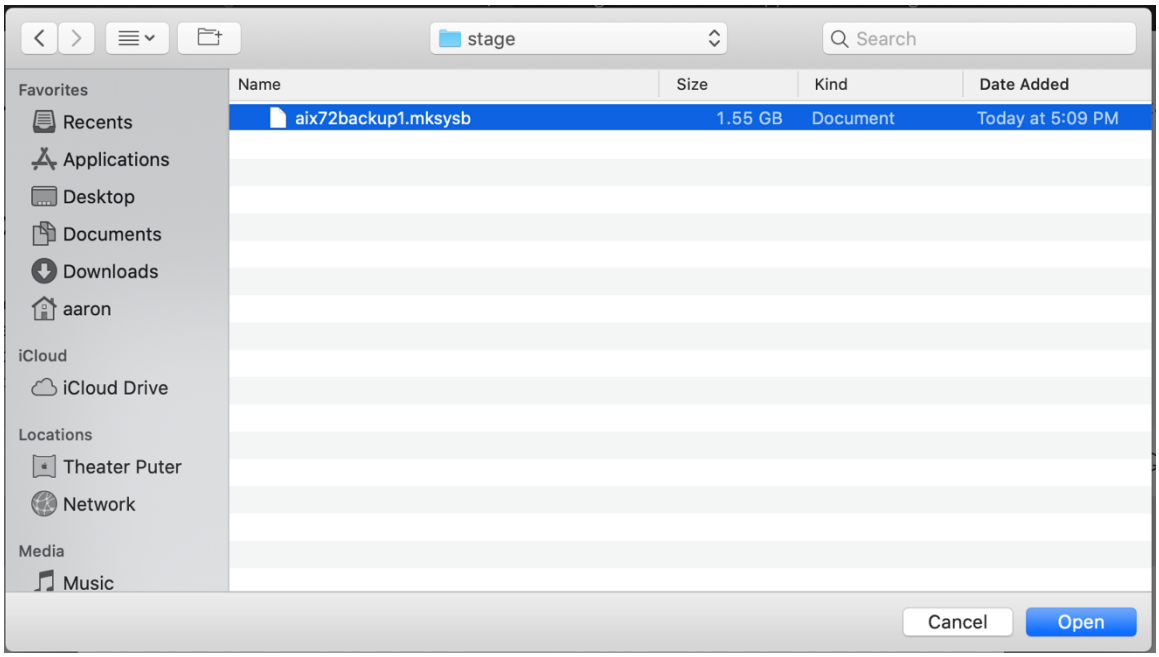
Now you can click the upload button found toward the right side of the GUI and select the option for Files.



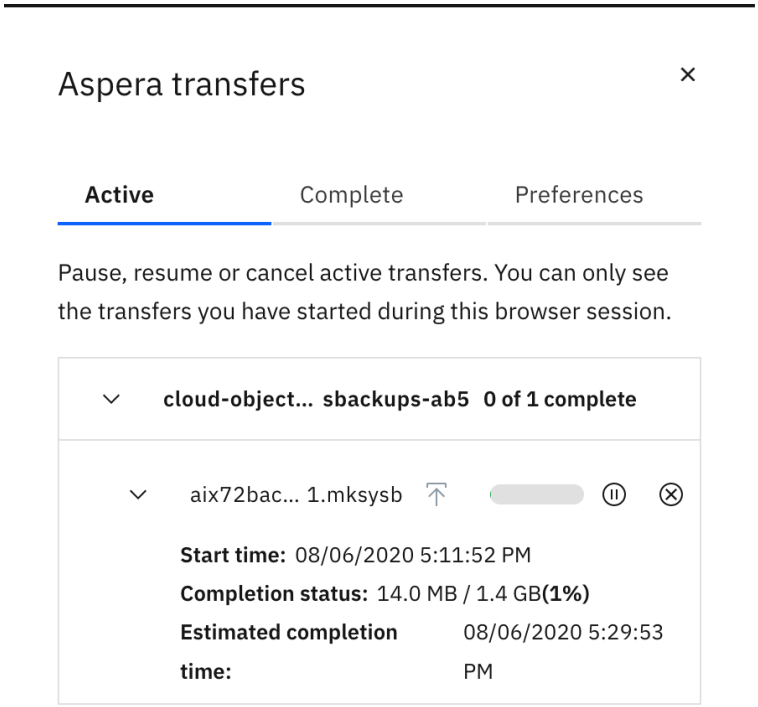
Then click the Select Files button on the pop up in the center of the page.



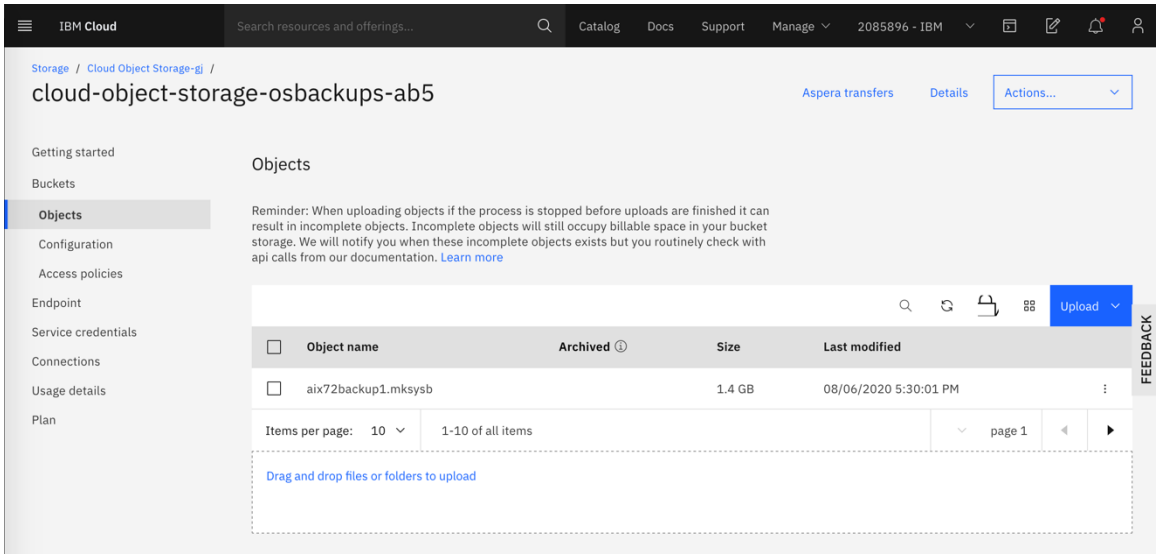
A file selection dialog from your workstation operating system will pop up and allow you to select the image that you plan to upload.



Once you've chose the file to upload. You can watch the status of the upload in the Aspera Transfers sidebar that will appear on the right.



Once the transfer is complete. You will be able to see the file you uploaded in the bucket.

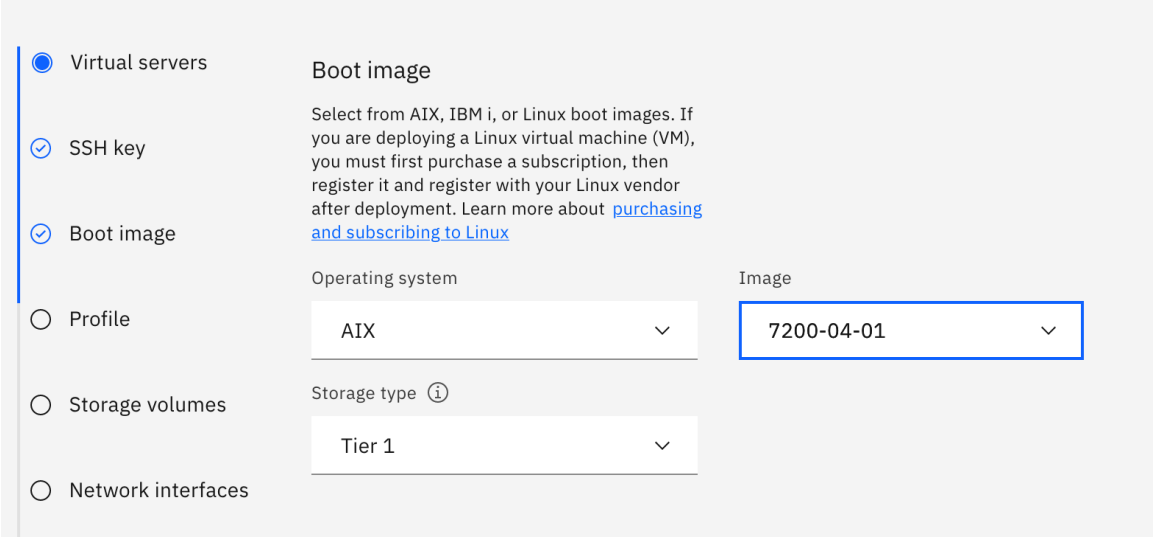


Restore the mksysb Backup to a New VSI

To restore a mksysb backup into a new Power VSI, first connect to your Linux staging VSI. List the available images in your Cloud Object Storage bucket and copy the desired image to the staging filesystem.

```
[root@labservices-osbackups-rhel-ab8 ~]# ls -l /cosbucket/
total 8882151
-rw-r--r--. 1 root root 9095321600 Jul 16 14:43 aix72backup1.mksysb
[root@labservices-osbackups-rhel-ab8 ~]# cp /cosbucket/aix72backup1.mksysb /stage/
[root@labservices-osbackups-rhel-ab8 ~]#
```

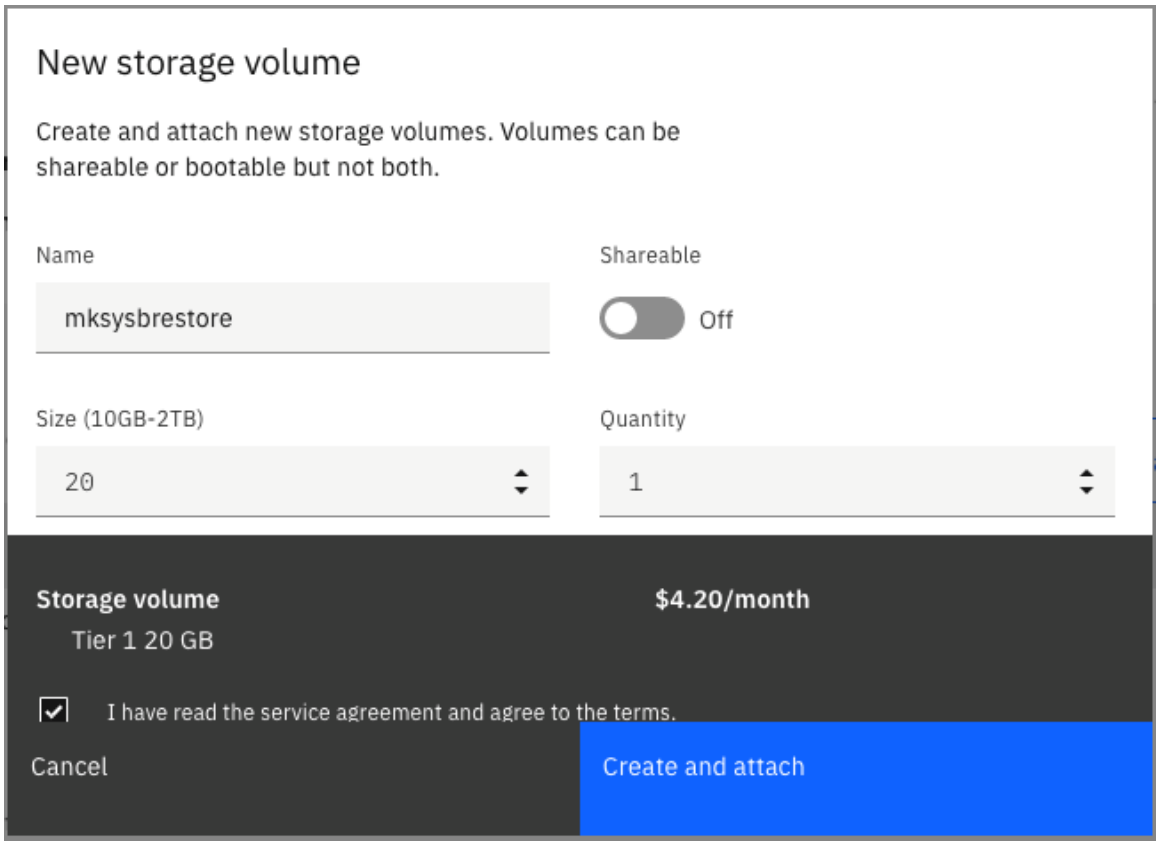
Now create a new Power VSI which will be the target for your mksysb restore. Choose one of the standard AIX boot images. This image will be used as a helper to perform the mksysb restore.



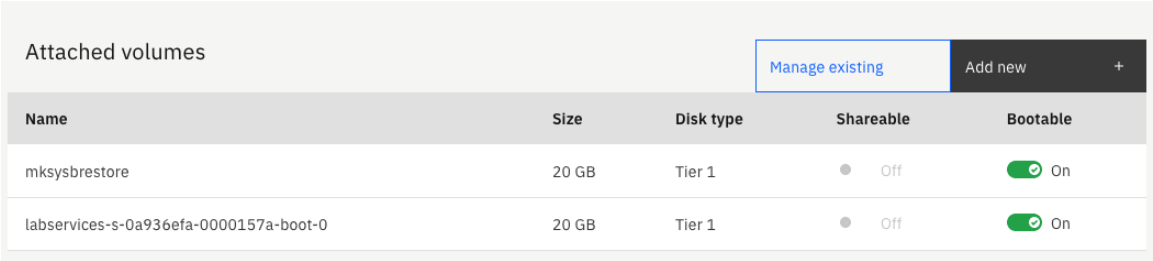
Once your new VSI is provisioned log in to it, create a mount point and mount the staging filesystem from your Linux VSI, which contains the mksysb that you will restore. Use ls to confirm the mksysb is available.

```
# mkdir /stage
# mount 10.72.253.136:/stage /stage
# ls -l /stage/aix72backup1.mksysb
-rw-r--r--    1 root     system   9095321600 Jul 16 14:32 /stage/aix72backup1.mksysb
#
```

Then return to the IBM Cloud GUI and add a new storage volume to your AIX VSI to use as a target for the mksysb restore.



Once the volume is attached, make sure the Bootable parameter is set to On. You may have to refresh your browser several times to see the status change.



Then run cfgmgr and use lspv to confirm the new disk is available.

```
# cfgmgr
# lspv
hdisk0          00f6db0af58e9775                    rootvg          active
hdisk1          none                                None
#
```

Next use the alt_disk_mksysb command to restore your mksysb on to the new disk.

```
# alt_disk_mksysb -c /dev/vty0 -d hdisk1 -m /stage/aix72backup1.mksysb
Restoring /image.data from mksysb image.
Checking disk sizes.
Creating cloned rootvg volume group and associated logical volumes.
Creating logical volume alt_hd5.
Creating logical volume alt_hd6.
Creating logical volume alt_hd8.
Creating logical volume alt_hd4.
Creating logical volume alt_hd2.
Creating logical volume alt_hd9var.
Creating logical volume alt_hd3.
Creating logical volume alt_hd1.
Creating logical volume alt_hd10opt.
Creating logical volume alt_hd11admin.
Creating logical volume alt_lg_dumplv.
Creating logical volume alt_livedump.
Creating logical volume alt_repo00.
Creating /alt_inst/ file system.
Creating /alt_inst/admin file system.
Creating /alt_inst/home file system.
Creating /alt_inst/opt file system.
Creating /alt_inst/tmp file system.
Creating /alt_inst/usr file system.
Creating /alt_inst/usr/sys/inst.images file system.
Creating /alt_inst/var file system.
Creating /alt_inst/var/adm/ras/livedump file system.
Restoring mksysb image to alternate disk(s).
Linking to 64bit kernel.
Changing logical volume names in volume group descriptor area.
Fixing LV control blocks...
forced unmount of /alt_inst/var/adm/ras/livedump
forced unmount of /alt_inst/var/adm/ras/livedump
forced unmount of /alt_inst/var
forced unmount of /alt_inst/var
forced unmount of /alt_inst/usr/sys/inst.images
forced unmount of /alt_inst/usr/sys/inst.images
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/usr
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/tmp
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/opt
forced unmount of /alt_inst/home
forced unmount of /alt_inst/home
forced unmount of /alt_inst/admin
forced unmount of /alt_inst/admin
forced unmount of /alt_inst
forced unmount of /alt_inst
Fixing file system superblocks...
Bootlist is set to the boot disk: hdisk0 blv=hd5
#
```

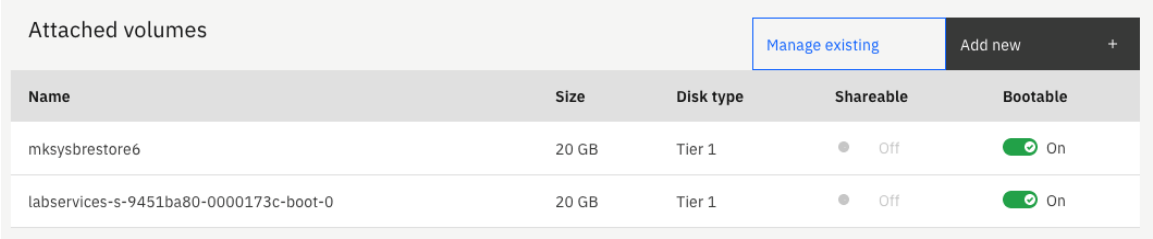
The bootlist is automatically modified to boot from the newly restored disk. Just reboot the VSI to begin using your restored image. This first boot may take some extra time while AIX is reconfigured to run in the new VSI.

Clean Up After a mksysb Restore

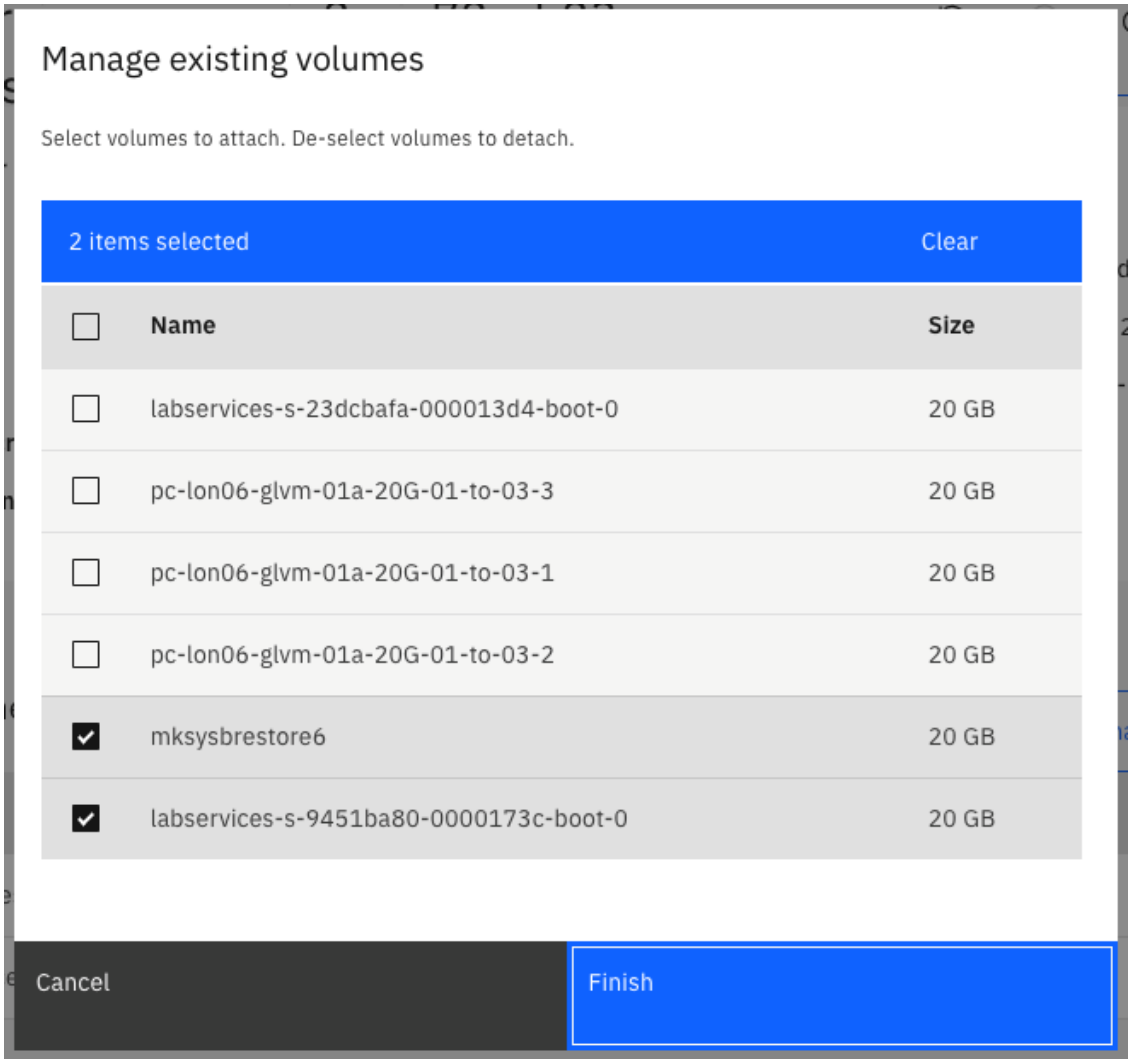
Once your new AIX VSI is running and you can log in, you can remove the original boot volume from the configuration. First use exportvg to remove the old rootvg then use rmdev to remove the original rootvg disk.

```
# lspv
hdisk0          00f6db0af58e9775                    old_rootvg
hdisk1          00c8d1607c6bb0be                    rootvg          active
# exportvg old_rootvg
# rmdev -Rdl hdisk0
hdisk0 deleted
#
```

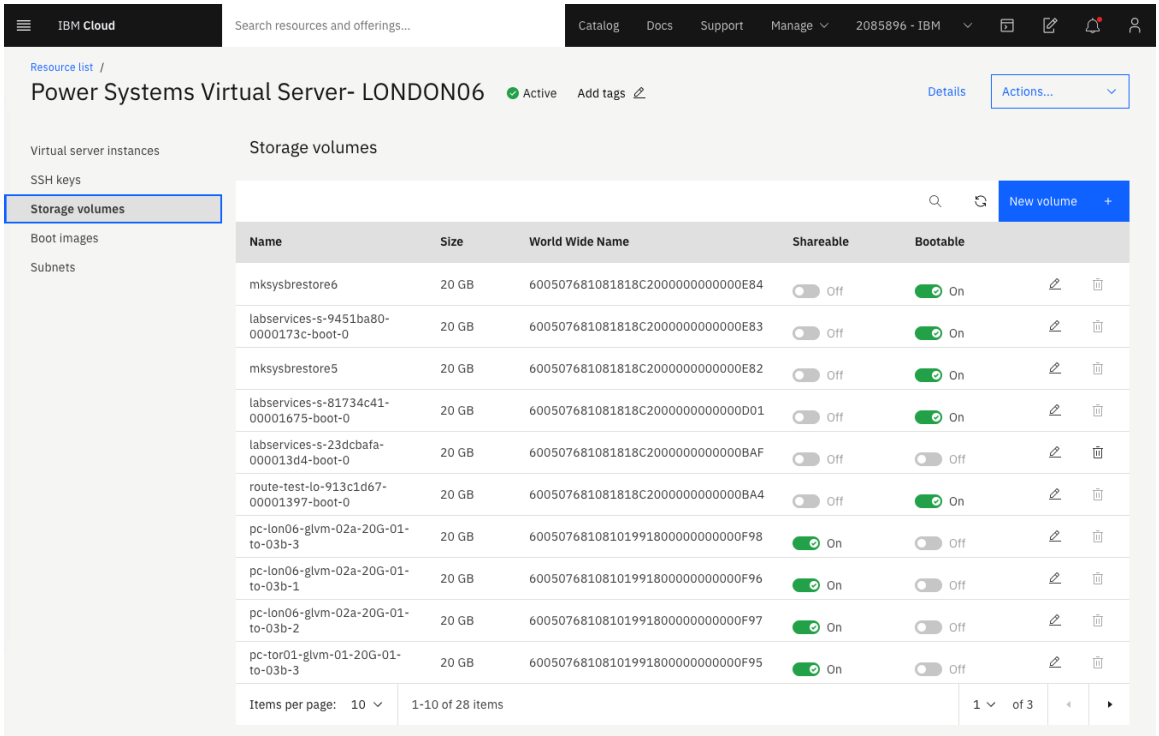
Then find the Attached Volumes section in your IBM Cloud GUI for your VSI. Click the Manage Existing button.



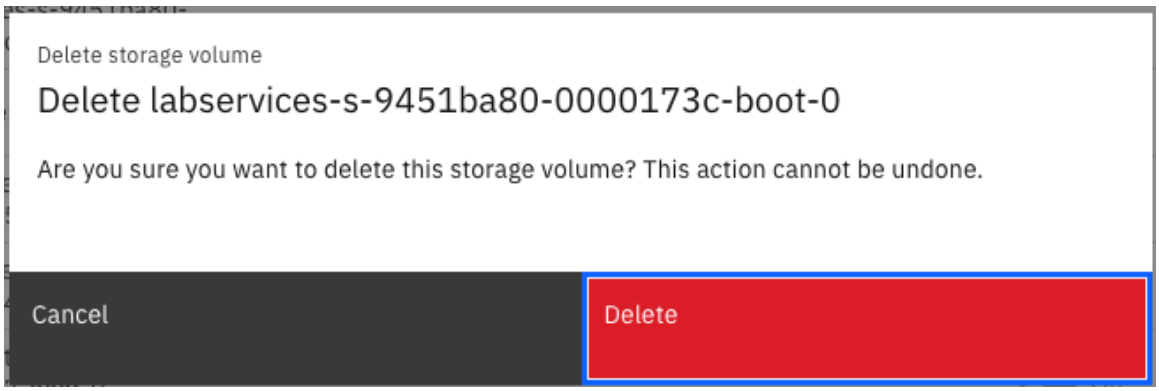
Then deselect the original boot volume, leaving your mksysb restore volume selected, and click Finish.



Finally, navigate to Storage Volumes in the left column. Locate your original boot volume and click the trash can icon on the right side of the listing to delete the volume.



Click the Delete button on the pop-up message to confirm.



This concludes the tutorial.