[Expert] Getting started

1. Connect to newly launched VM instance

- Using RDP client connect to VM instance with the default account name and initial password configured by the cloud provider.
- Change the default password to your own password.
- Optionally create additional users if needed.

2. Start the Crossmeta Drivers

- Launch Windows Command Prompt or PowerShell in Administrator mode.
- Navigate to Crossmeta installation directory *c:\program files\crossmeta* and run the script *service_crossmeta.cmd* as follows.

```
C:\Users\Administrator>cd "\Program Files\crossmeta"
C:\Program Files\crossmeta>service_crossmeta.cmd status
cxvfs is STOPPED cxfuse is STOPPED
cxnfs is STOPPED cxhfs is STOPPED
cxvfsmgr is STOPPED
C:\Program Files\crossmeta>service_crossmeta.cmd start
Starting Crossmeta kernel driver.
The CROSSMETA Manager service is starting.
The CROSSMETA Manager service was started successfully.
Running Crossmeta 2.0.2-RELEASE #240: Mon Apr 8 18:17:44 PDT 2019 on
WINNT/10.0_Build
C:\Program Files\crossmeta>
```

- Once Crossmeta drivers have loaded successfully the Crossmeta V: drive will be available for your use

On this page:

3. Configure your first object storage

objbaker is the program that provides access to object storage for the cloud gateway software. To start with the configuration you would need access key and secret generated by your provider, if you are using object storage in multi-cloud environment or if the compute engine was not authorized to perform any S3 object access operations.

AWS EC2

For AWS EC2 instance created with IAM role that authorizes S3 operations it may not be required to configure the accessId and accessKey within the software. Instead use the option --accessEC2IAM=ROLE option with objbaker. This will acquire S3 credentials from EC2 machine via IAM role.

- **Create access file with credentials**

  Create file v:\etc\.s3wasabi_passwd using notepad and add the access information in this format accessId:accessKey. The accessId could be account ID, and the accessKey is base64 encoded secret.

By starting filename with dot you would avoid Windows adding any .txt extension to the above file. Also make sure there are no extraneous characters or newlines in this file. This file is accessed by Crossmeta programs as /etc/.s3wasabi_passwd, while WIN32 programs access as v:\etc\.s3wasabi_passwd

- **Location of SSL certficate bundle**

  For SSL to work the curl library needs to know the location of its CA CERT bundle, which is installed in c:\program files\crossmeta. When you run the objbaker this information is provided with --cacert=ca-bundle.crt option . While running interactively since the startup directory is c:\program files\crossmeta this would work. But when the program is running as NT service the startup directory would be c:\windows\syswow64. In that case either copy ca-bundle.crt to c:\windows\syswow64 or specify full pathname to --cacert option.

- **List buckets with objbaker**

Before configuring object disk with Windows, make sure **objbaker** can list the buckets from your storage account. The baseURL is the most important parameter and should match with access URL of the cloud provider. You should use HTTPS for secure SSL access.

> **Info**
>
> For Azure baseURL will contain the <storage-account-name>.blob.core.windows.net
>
> The baseURL should end with slash character /
>
> The baseURL option can be skipped for Amazon AWS alone

Azure

```
C:\Program Files\crossmeta>objbacker.exe  --cacert=ca-bundle.crt --ssl  --accessFile=/etc/.
s3azure_passwd
--s3Type=azure --baseURL=https://zfstest.blob.core.windows.net/  --bucketCommand=list
```

AWS

```
C:\Program Files\crossmeta>objbacker.exe  --cacert=ca-bundle.crt  --ssl --accessFile=/etc/.
s3amazon_passwd
  --bucketCommand=list
```

Google Cloud

```
C:\Program Files\crossmeta>objbacker.exe --cacert=ca-bundle.crt --ssl  --accessFile=/etc/.
s3google_passwd
 --s3Type=google_s3  --baseURL=https://storage.googleapis.com/   --bucketCommand=list
```

Wasabi

```
C:\Program Files\crossmeta>objbacker.exe  --cacert=ca-bundle.crt --ssl  --accessFile=/etc/.
s3wasabi_passwd
 --baseURL=https://s3.us-west-1.wasabisys.com/   --bucketCommand=list
```

For troubleshooting add --debug-http to the above command to get detailed REST API output.

- **Configure object storage as VHD**

Before configuring object disk with Windows make sure the Crossmeta Fuse driver has started. If required start the driver as follows

FUSE

```
C:\Program Files\crossmeta>net start cxfuse
The CROSSMETA FUSE kernel mode driver service was started successfully.
```

- **Options to objbaker**

		Option
Logical size of VHDX up to 64TB allowed by Windows	10T	--size=10T
Block size of object files. Large block size reduces object access costs. This should match file system allocation size for optimal performance	1MB	--blockSize=1024K
Directory to use as mount point in Crossmeta drive	mnt	/mnt
No caching to prevent stale data		--directIo `--blockCacheSize=100`
Specify desired filename for object disk. This should have extension vhdx for Windows to use it		--filename=file.vhdx
S3 access credential file name. Some file in Crossmeta drive.	accessId:accessKey	--accessFile=/etc/. s3wasabi_passwd
S3 Type option to identify cloud object storage api. The default is amazon_s3. Mandatory option for Azure Blobs	azure google_s3 amazon_s3	--s3Type=azure
Region option. This option is mandatory for Amazon S3	us-west-1	--region=us-west-1

Access URL for the object storage. Must be specified for all S3 object provider other than Amazon
 	
Azure blobs contain the storage account name in the URL.
	zfstest is the storage account name for Azure	`--`
		`baseURL=https://zfstest.`
		`blob.core.windows.net/`

- Mount object storage bucket

  For Windows to make use of object storage bucket we have to intialize the object storage as VHDX image. VHDX is preferred over VHD because of inherent journal feature to protect against unsafe shutdown and also it can support up to 64TB for a single image. This step is performed in foreground interactively to make sure it is working before configuring it as service later. In the following example object storage bucket **smbtest1** is being mounted. Please make note of the -f option for foreground. Also for filename we chose as extension **.vhdx**

using bucket smbtest1 from Wasabi

```
C:\Program Files\crossmeta>objbacker.exe  --cacert=ca-bundle.crt  --accessFile=/etc/.s3wasabi_passwd
--authVersion=aws2
 --baseURL=https://s3.us-west-1.wasabisys.com/ --directIO --noAutoDetect --size=10T --timeout=30 --
blockCacheSize=100
--blockCacheThreads=8 --minWriteDelay=5000 --md5CacheSize=1000000 --ssl --blockSize=1024K -o
no_pagecache --filename=file.vhdx
 smbtest1  /mnt -f
objbacker.exe: auto-detection disabled; using configured block size 1m and file size 10t
 INFO: s3backer process 5312 for /mnt started
 INFO: mounting /mnt
```

using bucket disk1 from Azure

```
C:\Program Files\crossmeta>objbacker.exe  --cacert=ca-bundle.crt  --accessFile=/etc/.s3azure_passwd
--s3Type=azure
 --baseURL=https://storageaccount.blob.core.windows.net/ --directIO --noAutoDetect --size=10T --
timeout=30 --blockCacheSize=100
--blockCacheThreads=8 --minWriteDelay=5000 --md5CacheSize=1000000 --ssl --blockSize=1024K -o
no_pagecache --filename=file.vhdx
 disk1  /mnt -f
objbacker.exe: auto-detection disabled; using configured block size 1m and file size 10t
 INFO: s3backer process 5512 for /mnt started
 INFO: mounting /mnt
```

After the above command finishes the object storage is accessible as *V:\mnt\file.vhdx* as one giant file with logical capacity as 10TB.

qemu-img

```
C:\Program Files\qemu>dir v:\mnt
 Volume in drive V is vfsroot
 Volume Serial Number is 0122-B5C1

 Directory of v:\mnt

05/22/2019  07:48 PM    <DIR>          .
05/22/2019  07:48 PM    <DIR>          ..
05/22/2019  07:48 PM10,995,116,277,760 file.vhdx
05/22/2019  07:48 PM             1,422 stats
               2 File(s) 10,995,116,279,182 bytes
               2 Dir(s)       5,242,880 bytes free
```

- Initialize VHDX on the object storage bucket using **qemu-img** program available in c:\program files\qemu

qemu-img

```
cd c:\program files\qemu
C:\Program Files\qemu> qemu-img create -f vhdx v:\mnt\file.vhdx 10T
Formatting 'v:\mnt\file.vhdx', fmt=vhdx size=10995116277760 log_size=1048576 block_size=0
```

- Verify VHDX information

qemu-img

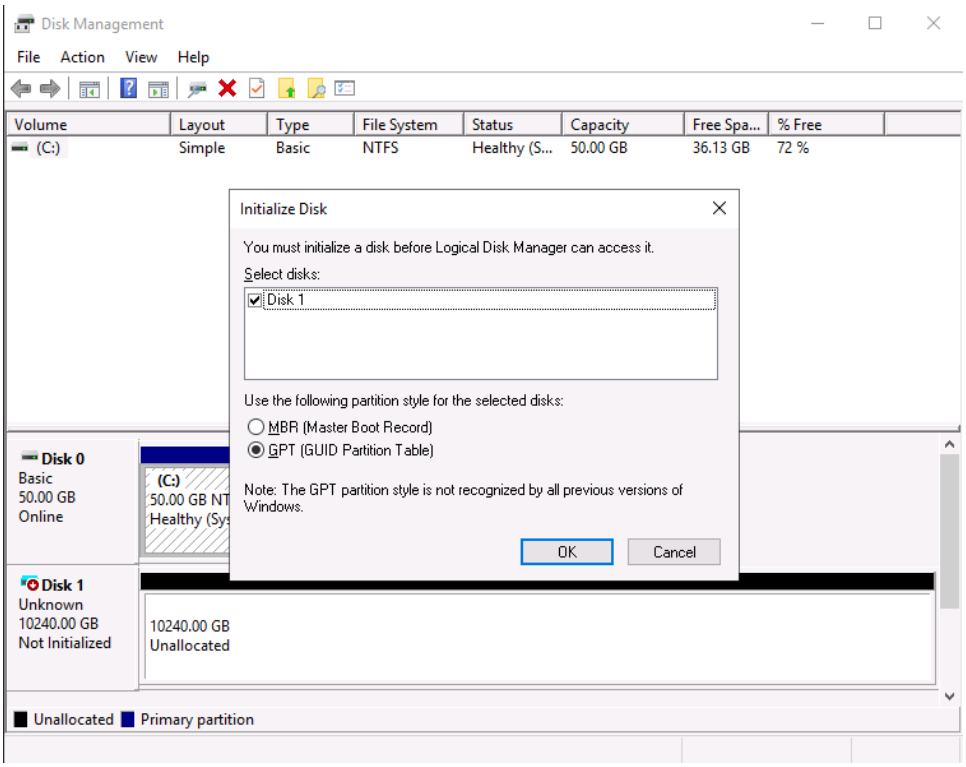
```
c:\Program Files\qemu> qemu-img info v:\mnt\file.vhdx
image: v:\mnt\file.vhdx
file format: vhdx
virtual size: 10T (10995116277760 bytes)
disk size: 10T
cluster_size: 33554432
```

- Activate VHDX file as virtual disk in Windows from Administrator PowerShell window

Attach VHDX

```
c:\users\Administrator> mount-diskimage  -imagepath v:\mnt\file.vhdx
```

- Launch diskmgmt.msc to initialize disk for the first time and assign a drive letter. You may create a file system within Disk Management GUI or from Command Prompt.



- Create a file system NTFS or ReFS. Use the actual drive letter that was assigned from the above command and allocation size of 1MB is chosen.

Format

```
c:\users\Administrator> format /fs:ntfs /A:1M  /q  F:
```

NTFS allocation size

Windows 2019 supports allocation size of 1MB for NTFS file systems while earlier version limited to 64KB only. This means NTFS created with 1MB allocation size may not be recognized by older Windows.

- **Detach the virtual disk**

Detach VHDX

```
c:\users\Administrator> dismount-diskimage  -imagepath v:\mnt\file.vhdx
```

- **Stop the objbaker program to proceed configuration as service**

From Desktop Crossmeta command shell running as administrator

Stop objbaker

```
V:\>mount
/dev/loop0 on / type ufs (local, read-only)
devfs on /dev type devfs (local)
\fuse.chan3596 on /mnt type fuse

V:\> umount /mnt
```

The above command would stop the objbacker program that is running inside another Command Prompt Window.

4. Register as NT service

- **Create service using sc.exe**

If everything worked out fine so far we are ready to make this object disk as service now. Pick a service name to start this object disk as NT service program automatically from now on. In this example we register objdisk1 as NT service program.

In the following command line for sc.exe please note that binPath= and displayName= has no space character before = but space character after =

Create service

```
C:\Program Files\crossmeta>sc  create  objdisk1  start= delayed-auto binpath="\"c:\program
files\crossmeta\objbacker.exe\" --cacert=\"c:\program files\crossmeta\ca-bundle.crt\" --accessFile=
/etc/.s3wasabi_passwd  --baseURL=https://s3.us-west-1.wasabisys.com/ --directIO  --noAutoDetect  --
size=10T --timeout=30 --blockCacheSize=100 --blockCacheThreads=8 --minWriteDelay=5000 --
md5CacheSize=1000000 --ssl --blockSize=1024K -o no_pagecache --filename=file.vhdx smbtest1  /mnt"
displayName= "My first object disk smbtest1 from Wasabi" depend= cxfuse
[SC] CreateService SUCCESS


C:\Program Files\crossmeta>sc qc objdisk1
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: objdisk1
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START  (DELAYED)
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : "c:\program files\crossmeta\objbacker.exe" --cacert="c:\program
files\crossmeta\ca-bundle.crt" --accessFile=/etc/.s3wasabi_passwd  --baseURL=https://s3.us-west-1.
wasabisys.com/ --directIO  --noAutoDetect  --size=10T --timeout=30 --blockCacheSize=100 --
blockCacheThreads=8 --minWriteDelay=5000 --md5CacheSize=1000000 --ssl --blockSize=1024K -o
no_pagecache --filename=file.vhdx smbtest1  /mnt
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : My first object disk smbtest1 from Wasabi
        DEPENDENCIES       : cxfuse
        SERVICE_START_NAME : LocalSystem
```

- **Starting the objdisk1 service**

Start

```
sc start objdisk1
powershell  mount-diskimage  -imagepath v:\mnt\file.vhdx
```

- **Stopping the objdisk service**

From Desktop Crossmeta command shell running as administrator

Stop

```
v:\> umount /mnt

v:\> sc stop objdisk1
```

5. Automatic Startup

- **Create startup script**

To configure object disk every time the system boots up create a command batch file in **c:\program files\crossmeta\s3disks.cmd** and add the following to file assuming *objdisk1* being the configured service name and */mnt* as the mount point.

s3disks.cmd file

```
net start objdisk1
powershell mount-diskimage -imagepath v:\mnt\file.vhdx
```

- **Schedule the above as task**

To run **s3disks.cmd** every time the Windows starts use *schtasks.exe*

schtasks.exe

```
schtasks.exe /create /tn "Start Cloud Disks" /ru SYSTEM /Sc ONSTART /tr
"\"C:\program files\s3disks.cmd\""
```

Note how quote characters were escaped while specifying the command line argument to schtasks.exe, since the directory pathname contains space character.

Related pages