



テクニカルホワイトペーパー

**Veritas InfoScale Enterprise 7.4.1 for RHEL on Azure  
ストレージメンテナンス手順**

---

2020年5月

ベリタステクノロジーズ合同会社  
テクノロジーセールス本部

**VERITAS™**

The truth in information.

## 免責事項

ベリタステクノロジーズ合同会社は、この文書の著作権を留保します。また、記載された内容の無謬性を保証しません。Veritas InfoScale は将来に渡って仕様を変更する可能性を常に含み、これらは予告なく行われることもあります。なお、当ドキュメントの内容は参考資料として、読者の責任において管理/配布されるようお願いいたします。

## 目次

免責事項 .....	2
1. はじめに .....	4
本書の目的 .....	4
2. 非クラスター構成でのデータ領域拡張手順 .....	5
前提となるシステム構成 .....	5
既存のブロックストレージを拡張して、ボリュームを拡張するパターン .....	6
新規にブロックストレージを追加して、ボリュームを拡張するパターン .....	10
3. 共有ディスク型クラスター構成でのデータ領域拡張手順 .....	17
前提となるシステム構成 .....	17
既存のブロックストレージを拡張して、ボリュームを拡張するパターン .....	18
新規にブロックストレージを追加して、ボリュームを拡張するパターン .....	24
4. FSS を用いた仮想ミラー型クラスター構成でのデータ領域拡張手順 .....	33
本章の前提となるシステム構成 .....	33
既存のブロックストレージを拡張して、ボリュームを拡張するパターン .....	34
新規にブロックストレージを追加して、ボリュームを拡張するパターン .....	41
5. VVR を用いたレプリケーション型クラスター構成でのデータ領域拡張手順 .....	50
本章の前提となるシステム構成 .....	50
既存のブロックストレージを拡張して、ボリュームを拡張するパターン .....	51
新規にブロックストレージを追加して、ボリュームを拡張するパターン .....	58

## 1. はじめに

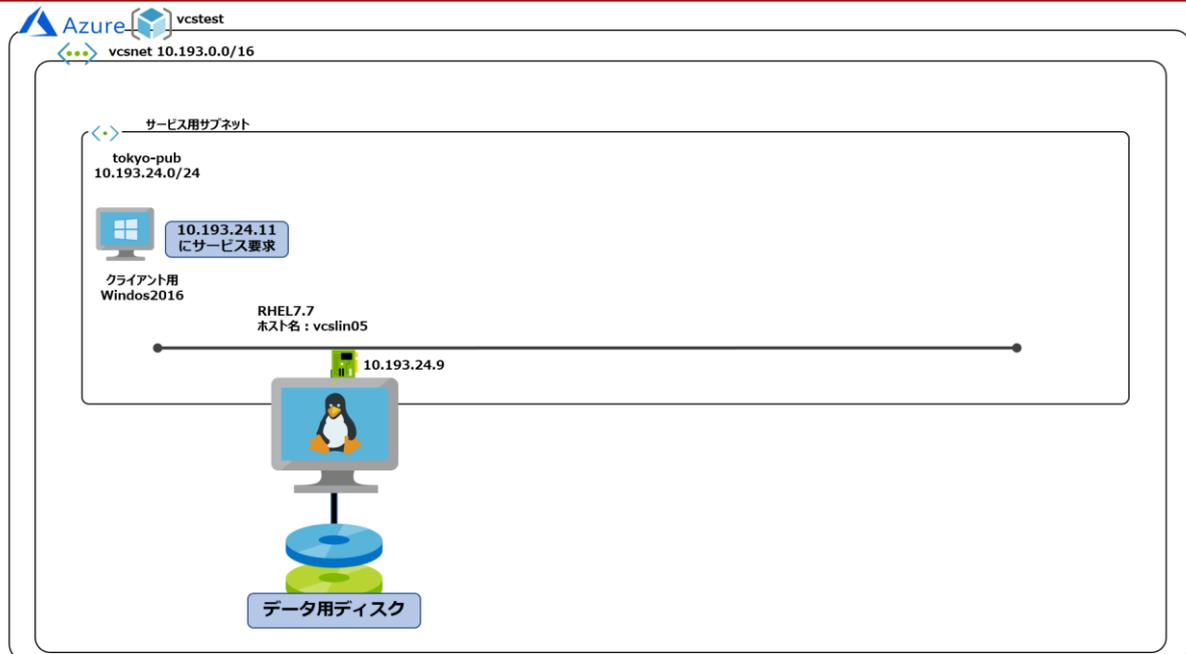
### 本書の目的

本書は、Azure 上の RHEL に InfoScale Enterprise 7.4.1 を構築した環境における、データ領域の拡張手順の理解を目的に作成されています。本書は、非クラスター構成、共有ディスク型クラスター構成、FSS による仮想ミラーリング構成、VVR によるレプリケーション構成、の 4 種類の構成をカバーしています。

## 2. 非クラスター構成でのデータ領域拡張手順

この章では、シングルノード構成における InfoScale を用いたデータ領域を拡張するシナリオを説明します。

### 前提となるシステム構成

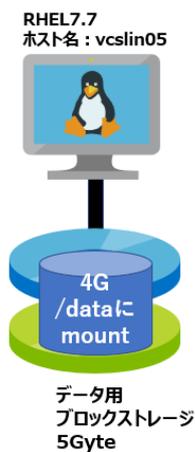


前提となる構成イメージ

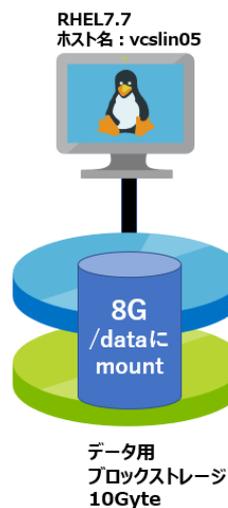
## 既存のブロックストレージを拡張して、ボリュームを拡張するパターン

拡張前と後の構成は以下です。拡張前はデータ用として 5Gbyte のブロックストレージが接続されており、4Gbyte のデータ用ボリューム：vol01 があります。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、/data にマウントされています。この状況下で、/data の容量を 8Gbyte に拡張する要件が発生したと想定します。要件に応じて、データ用ブロックストレージが 10Gbyte に拡張され、データ用ボリュームとその上位のベリタスファイルシステムが同時に 8Gbyte まで拡張されます。

### 拡張前



### 拡張後



ここからは、実際の拡張作業を説明します。拡張前に、InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

まず、VxVM の管理下にあるディスクを確認します。3 本のディスクが確認されていますが、VxVM の管理下である事を示す「STATUS が Online」のディスクは 10-193-24-8\_disk\_7 の 1 つだけです。

```
# vxdisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
10-193-24-8_disk_2	auto:LVM	-	-	LVM
10-193-24-8_disk_6	auto:none	-	-	online invalid
10-193-24-8_disk_7	auto:cdsdisk	10-193-24-8_disk_0	datadg	online

次に、10-193-24-8\_disk\_7 が OS からどのような名前で見られているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
```

SUBPATH	DANAME	DMNAME	GROUP	STATE
sda	10-193-24-8_disk_2	-	-	ENABLED
sdb	10-193-24-8_disk_6	-	-	ENABLED
sdc	10-193-24-8_disk_7	10-193-24-8_disk_0	datadg	ENABLED

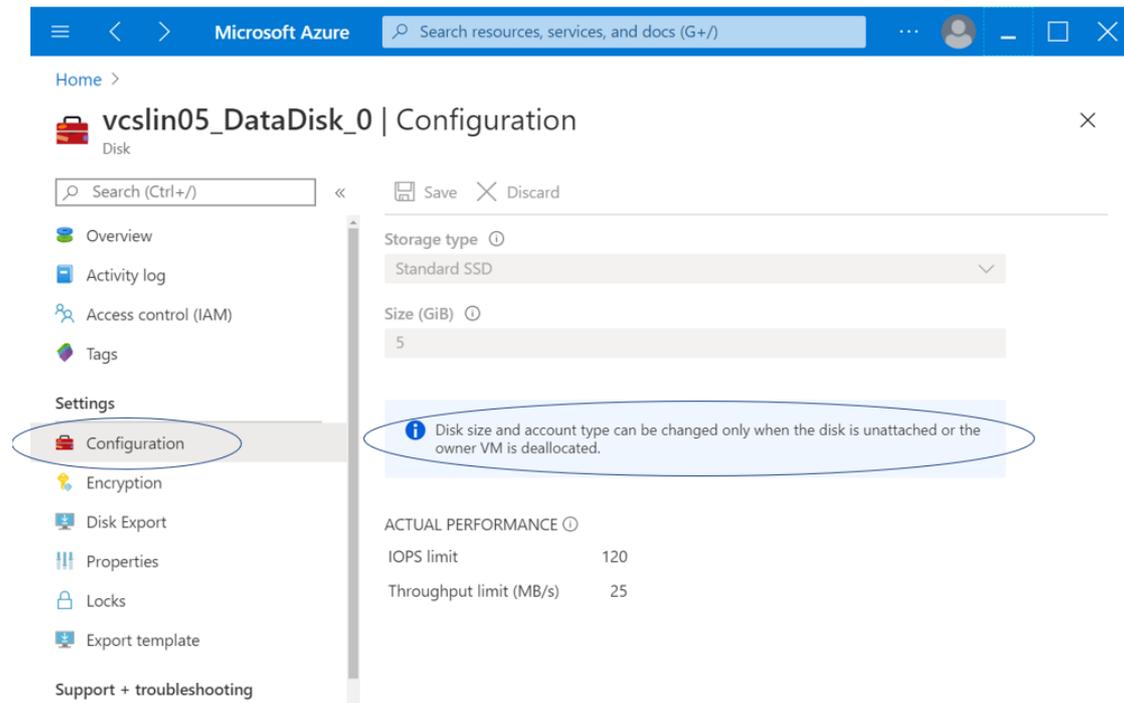
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    5G  0 disk
├─sdc3       8:35  0    5G  0 part
└─sdc8       8:40  0    5G  0 part
```

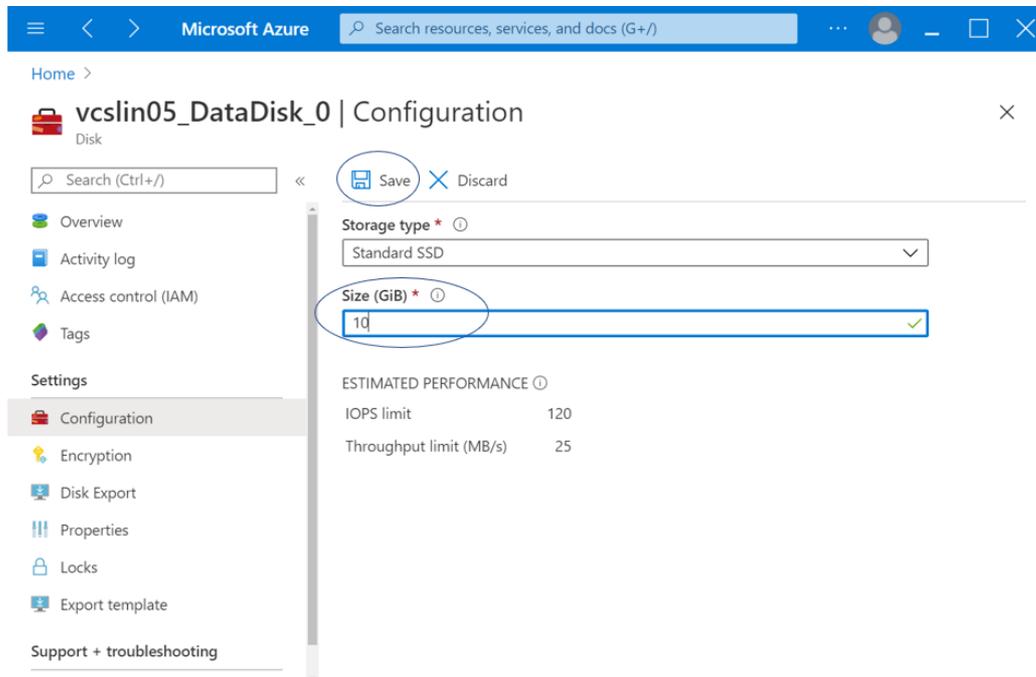
念の為、拡張対象のファイルシステム：/data のサイズを確認します。5Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 4.0G 20M 3.8G 1% /data
```

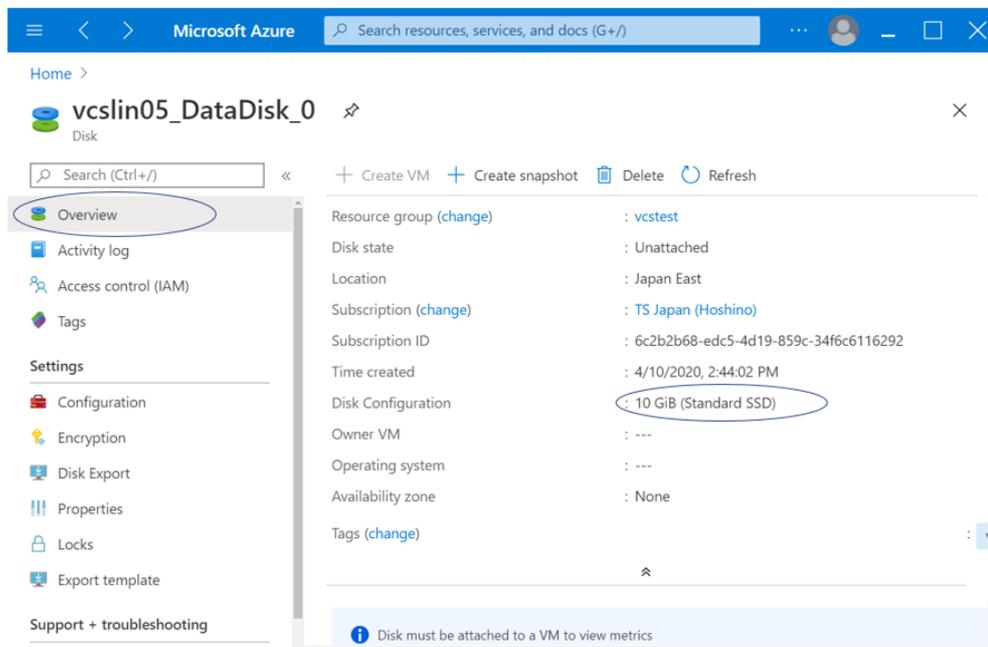
次に、Azure のコンソールで、ブロックストレージの容量を変更します。下記は、拡張対象のブロックストレージを指定し「configuration」を表示した画面です。「Disk size and account type can be changed only when...」の記述の通り、ブロックストレージのサイズ変更は、仮想マシンを停止するか、仮想マシンからブロックストレージを切り離さないとできません。この場合は、仮想マシンを停止します。



仮想マシンを停止後、再度 Azure のコンソールで同じ画面を開くと、「Disk size and account type can be changed only when...」の記述が消え、サイズ変更が可能になります。拡張後のサイズを指定し「save」をクリックしてください。



ブロックストレージの「Overview」の画面で、正しくサイズ拡張されたのを確認したら、仮想マシンを再起動してください。



仮想マシンが立ち上がったら、lsblk コマンドを用い sdc のサイズを確認します。想定通り 10Gbyte に拡張されています。

```
# lsblk | grep sdc
sdc          8:32  0    10G  0 disk
├─sdc3       8:35  0    10G  0 part
└─sdc8       8:40  0    10G  0 part
```

次に、vxprint コマンドで VxVM が認識しているデータ用ブロックストレージのサイズを確認します。まだ、10411776 ブロック、すなわち 5Gbyte で認識されています。

```
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -

dm 10-193-24-8_disk_0 10-193-24-8_disk_7 10411776 -        -        -

v vol01      fsgen     ENABLED 8388608 -        ACTIVE  -        -
pl vol01-01  vol01     ENABLED 8388608 -        ACTIVE  -        -
sd 10-193-24-8_disk_0-01 vol01-01 ENABLED 8388608 0        -        -        -
```

vxdisk コマンドに、ディスクグループ名である datadg と、再認識対象ディスク名の 10-193-24-8\_disk\_0 を指定して、ディスクサイズを再認識させます。

```
# vxdisk -f -g datadg resize 10-193-24-8_disk_0
```

vxprint で確認すると、確かに 10-193-24-8\_disk\_0 のサイズが 20904976 ブロック、すなわち 10Gbyte と認識されています。

```
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -

dm 10-193-24-8_disk_0 10-193-24-8_disk_7 20904976 -        -        -

v vol01      fsgen     ENABLED 8388608 -        ACTIVE  -        -
pl vol01-01  vol01     ENABLED 8388608 -        ACTIVE  -        -
sd 10-193-24-8_disk_0-01 vol01-01 ENABLED 8388608 0        -        -        -
```

次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 8Gbyte に拡張します。ここで指定するのは、マウントポイント名ではなく、ディスクグループ名（この場合は datadg）とボリューム名（vol01）である事に注意してください。

```
# vxresize -g datadg vol01 8g
```

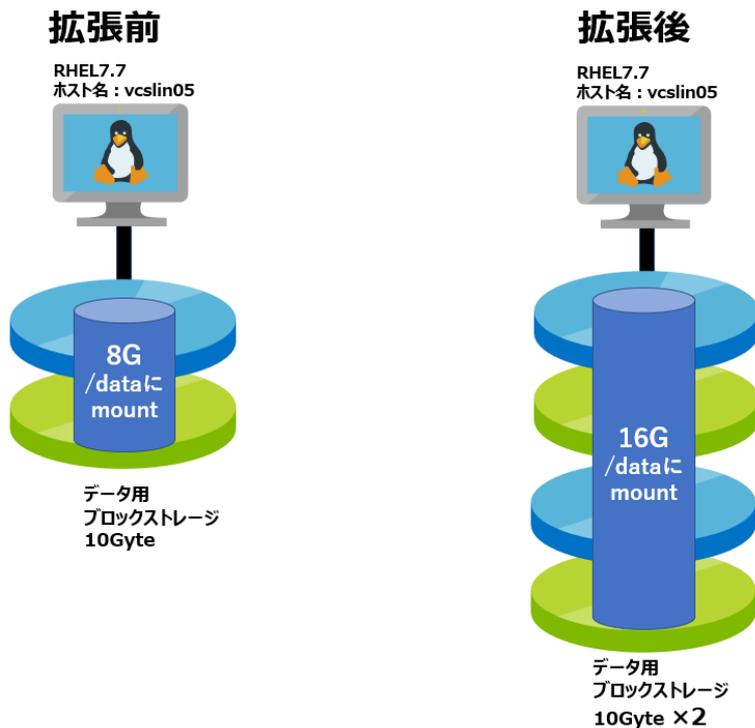
df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01  8.0G  21M  7.5G  1% /data
```

これで作業は終了です。

## 新規にブロックストレージを追加して、ボリュームを拡張するパターン

拡張前と後の構成は以下です。拡張前はデータ用として 10Gbyte のブロックストレージが接続されており、8Gbyte のデータ用ボリューム：vol01 があります。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、/data にマウントされています。この状況下で、/data の容量を 16Gbyte に拡張する要件が発生したと想定します。要件に応じて、10Gbyte のデータ用ブロックストレージが新規で追加され、それを使用してボリュームとその上位のベリタスファイルシステムが同時に 16Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。拡張前に、InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

まず、VxVM の管理下にあるディスクを確認します。3 本のディスクが確認されていますが、VxVM の管理かである事を示す「STATUS が Online」のディスクは 10-193-24-8\_disk\_7 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
```

次に、10-193-24-8\_disk\_7 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE          ENABLED
sda              10-193-24-8_disk_2 -             -             ENABLED
sdb              10-193-24-8_disk_6 -             -             ENABLED
sdc              10-193-24-8_disk_7 10-193-24-8_disk_0 datadg        ENABLED
```

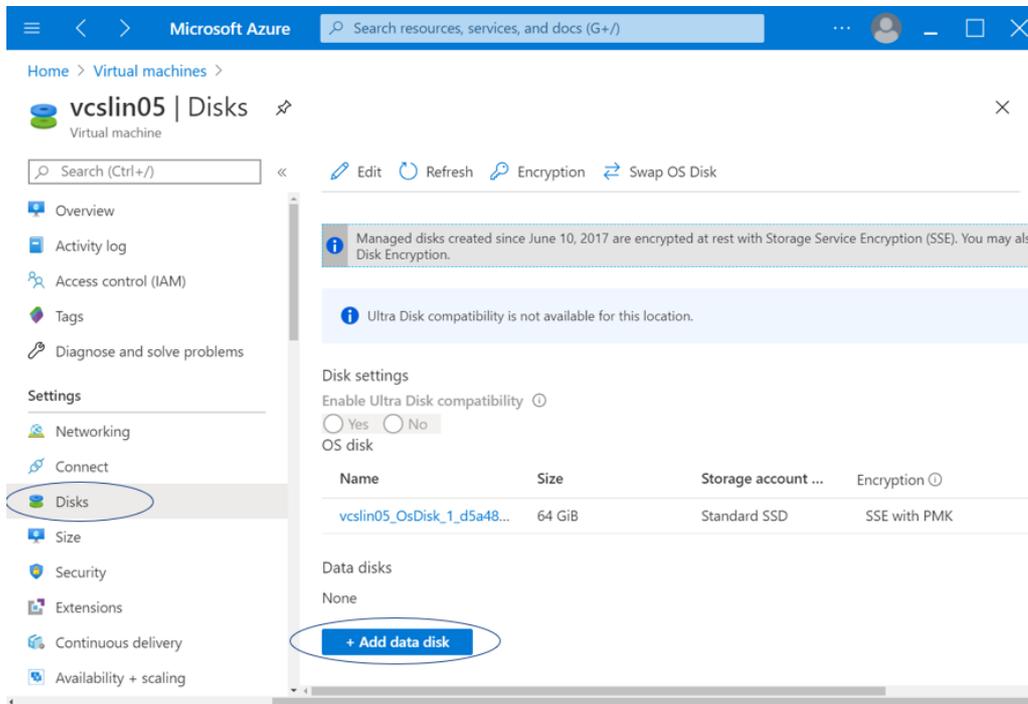
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。10Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    10G  0 disk
├─sdc3       8:35  0    10G  0 part
└─sdc8       8:40  0    10G  0 part
```

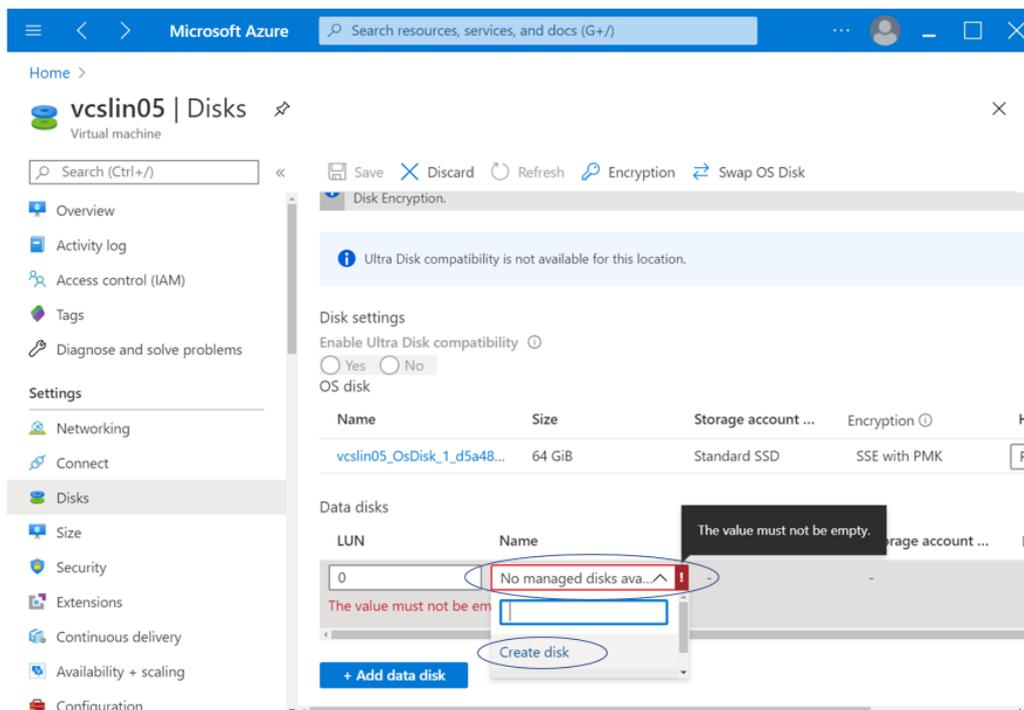
念の為、拡張対象のファイルシステム：/data のサイズを確認します。8Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 8.0G 20M 3.8G 1% /data
```

次に、Azure のコンソールで、ブロックストレージを追加します。ブロックストレージの追加は、仮想マシンを停止しないとできないので、インスタンスを停止してから下記の作業を行ってください。対象の仮想マシンのメニューで「Disk」を指定し、「Add data disk」をクリックしてください。



「Data disks」の欄の「Name」の下のプルダウンメニューを開くと「Create disk」が表示されますので、これをクリックしてください。



「Disk name」と「Resource group」を入力したら「Change size」をクリックしてください。

Home > vcslin05 | Disks >

### Create a managed disk

Create a new disk to store applications and data on your VM. Disk pricing varies based on factors including disk size, storage type, and number of transactions.

**Disk name \***

**Resource group \***

[Create new](#)

**Location**  
Japan East

**Availability zone**  
None

**Source type**  
None

**Size \***  
1024 GiB  
Standard SSD  
[Change size](#)

**Encryption type \***  
(Default) Encryption at-rest with a platform-managed key

[Create](#)

サイズを入力して「OK」をクリックしてください。

Home > v

### Select a disk size

128 GiB	\$10	500	60
256 GiB	\$15	500	60
512 GiB	\$20	500	60
1024 GiB	\$30	500	60
2048 GiB	\$40	500	60
4096 GiB	\$50	500	60
8192 GiB	\$60	1300	300
16384 GiB	\$70	2000	500
32767 GiB	\$80	2000	500

**Size \***

**1024 GiB**  
Standard SSD  
[Change size](#)

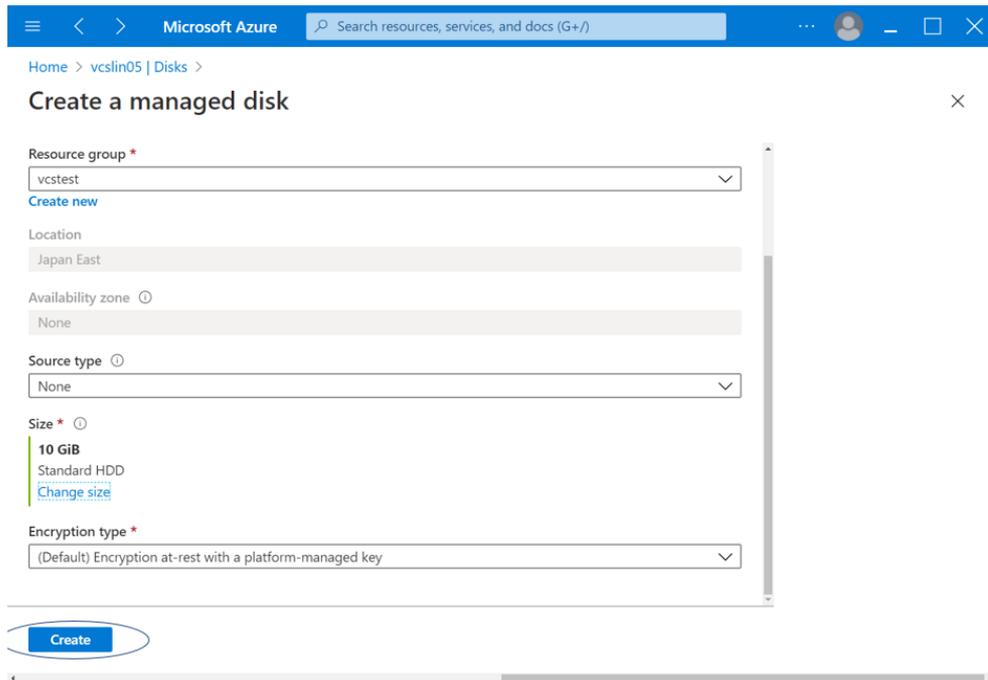
**Encryption type \***  
(Default) Encryption at-rest with a platform-managed key

**Create a custom size**  
Enter the size of the disk you would like to create. You will be charged the same rate for your provisioned disk, regardless of how much of the disk space is being used. For example, a 200 GiB disk is provisioned on a 256 GiB disk, so you would be billed for the 256 GiB provisioned.

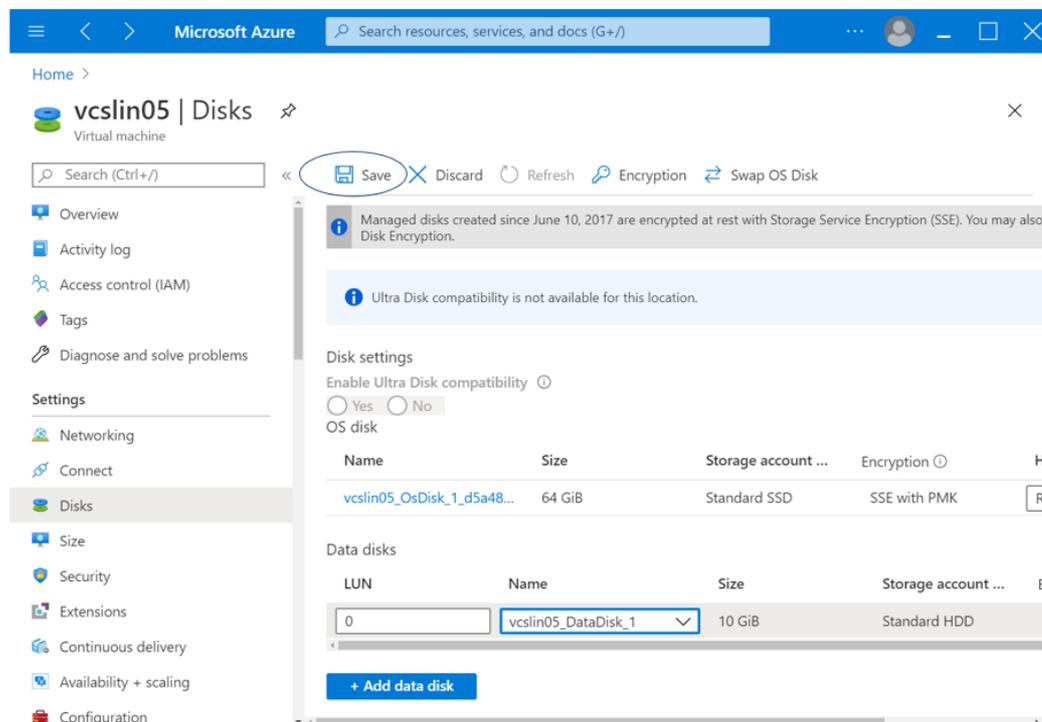
**Custom disk size (GiB) \***

[Create](#) [OK](#)

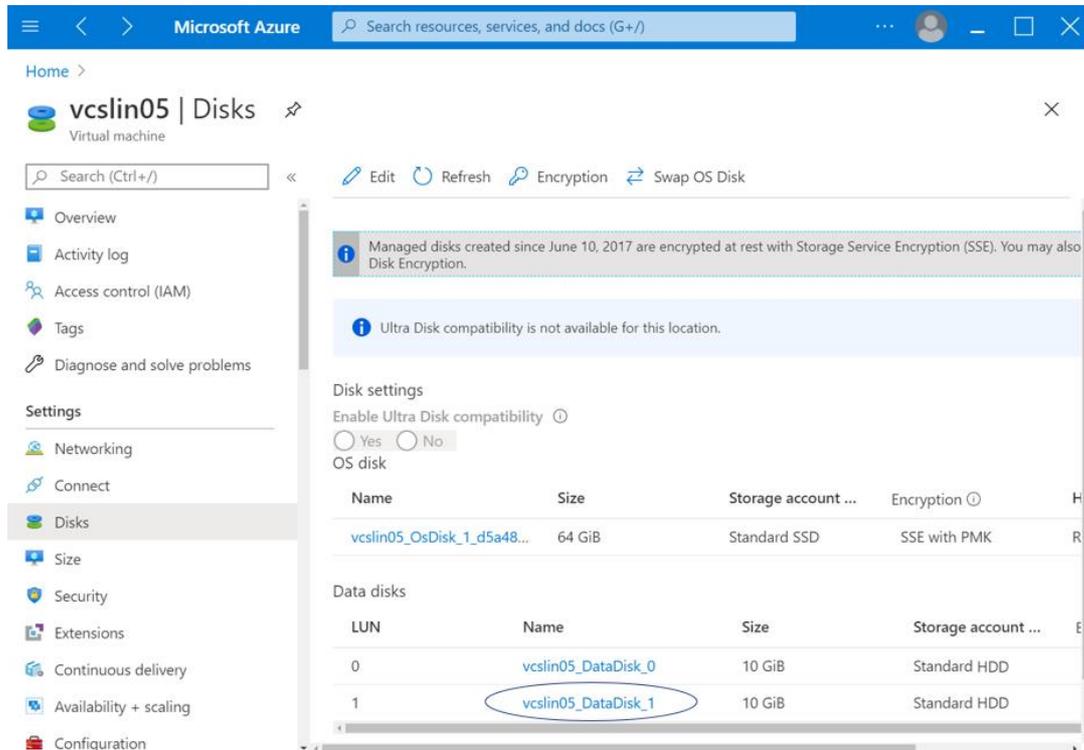
内容を確認して「Create」をクリックしてください。



作成したブロックストレージが「Data disks」欄に表示されますので、Name やサイズなどの内容を確認の上「Save」をクリックしてください。



再度、対象の仮想マシンのメニューで「Disk」を指定すると、作成したブロックストレージが「Data disks」欄に表示されます。確認後、仮想マシンを立ち上げてください。



仮想マシンが立ち上がったら、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda      8:0    0   64G  0 disk
├── sda1  8:1    0   500M  0 part /boot/efi
├── sda2  8:2    0   500M  0 part /boot
├── sda3  8:3    0    2M   0 part
└── sda4  8:4    0    63G  0 part
sdb      8:16   0   50G  0 disk
└── sdb1  8:17   0   50G  0 part /mnt/resource
sdc      8:32   0   10G  0 disk
├── sdc3  8:35   0   10G  0 part
└── sdc8  8:40   0   10G  0 part
sdd      8:48   0   10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前で認識されているか確認します。10-193-24-8\_disk\_8 という名前で認識されている事がわかりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              10-193-24-8_disk_2 -                -              ENABLED
sdb              10-193-24-8_disk_6 -                -              ENABLED
sdc              10-193-24-8_disk_7 10-193-24-8_disk_0 datadg        ENABLED
sdd              10-193-24-8_disk_8 -                -              ENABLED
```

vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:none     -             -             online invalid
```

次に、vxdisksetup コマンドで 10-193-24-8\_disk\_8 をイニシャライズします。

```
# vxdisksetup -i 10-193-24-8_disk_8
```

再度 vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 の STATUS は online に変わり、VxVM の管理下になった事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:cdsdisk  -             -             online
```

vxchg コマンドで、既存のディスクグループ：datadg に 10-193-24-8\_disk\_8 を加えます。vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 が datadg に所属した事が確認できます。これで、追加されたディスク：10-193-24-8\_disk\_8 を用いて既存のボリューム：vol01 を拡張する準備が整いました。

```
# vxchg -g datadg adddisk 10-193-24-8_disk_8
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:cdsdisk  10-193-24-8_disk_8 datadg        online
```

次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 16Gbyte に拡張します。ここで指定するのは、マウントポイント名ではなく、ディスクグループ名（この場合は datadg）とボリューム名（vol01）である事に注意してください。

```
# vxresize -g datadg vol01 16g
```

df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 16G 23M 15G 1% /data
```

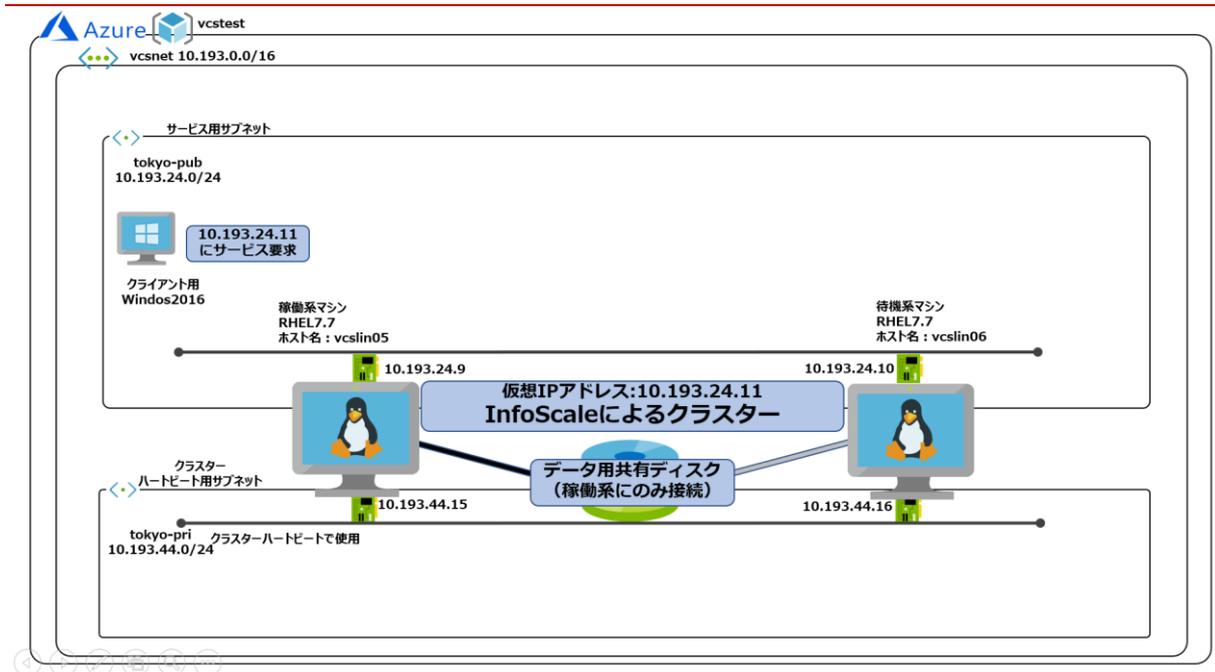
これで作業は終了です。

### 3. 共有ディスク型クラスター構成でのデータ領域拡張手順

この章では、共有ディスク型クラスター構成における InfoScale を用いたデータ領域を拡張するシナリオを説明します。尚、Azure 上で InfoScale を用いて共有ディスク型クラスターを構築する手順については、ベリタスより詳細が公開されています。詳しくは

[https://www.veritas.com/support/en\\_US/doc/InfoScale\\_7.4.1\\_RHEL\\_on\\_Azure\\_deploy\\_PrivateIP](https://www.veritas.com/support/en_US/doc/InfoScale_7.4.1_RHEL_on_Azure_deploy_PrivateIP) を参照してください。

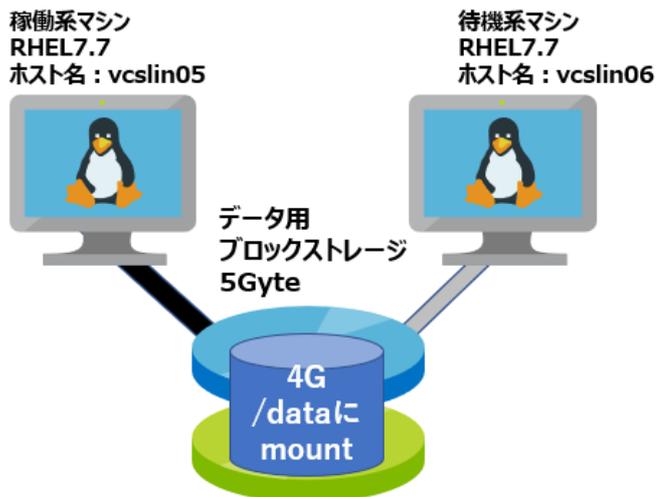
#### 前提となるシステム構成



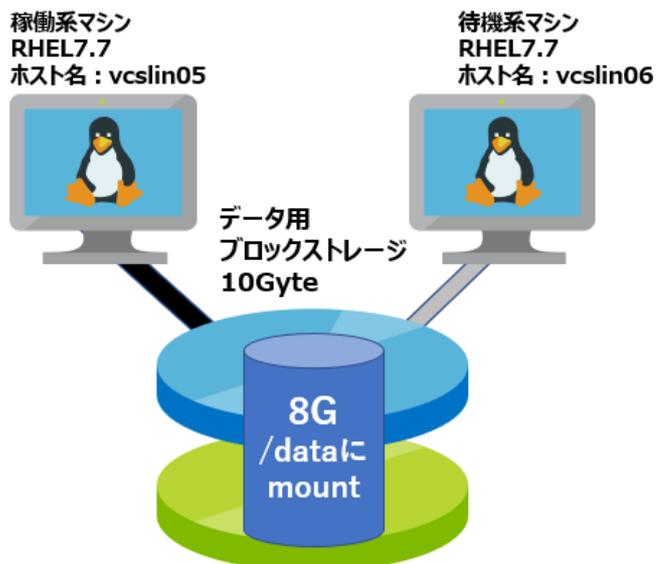
前提となる構成イメージ

## 既存のブロックストレージを拡張して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 5Gbyte のブロックストレージが共有接続されており（ただし、アクセスできるのは稼働系のみ）、その中に 4Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、稼働系の /data にマウントされます。この状況下で、/data の容量を 8Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。ブロックストレージが 10Gbyte に拡張され、データ用ボリュームとその上位のベリタスファイルシステムが同時に 8Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。この作業はクラスターの稼働系で行います。まず、拡張前に InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

まず、VxVM の管理下にあるディスクを確認します。3 本のディスクが確認されていますが、VxVM の管理下である事を示す「STATUS が Online」のディスクは 10-193-24-8\_disk\_7 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
```

次に、10-193-24-8\_disk\_7 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              10-193-24-8_disk_2 -             -             ENABLED
sdb              10-193-24-8_disk_6 -             -             ENABLED
sdc              10-193-24-8_disk_7 10-193-24-8_disk_0 datadg        ENABLED
```

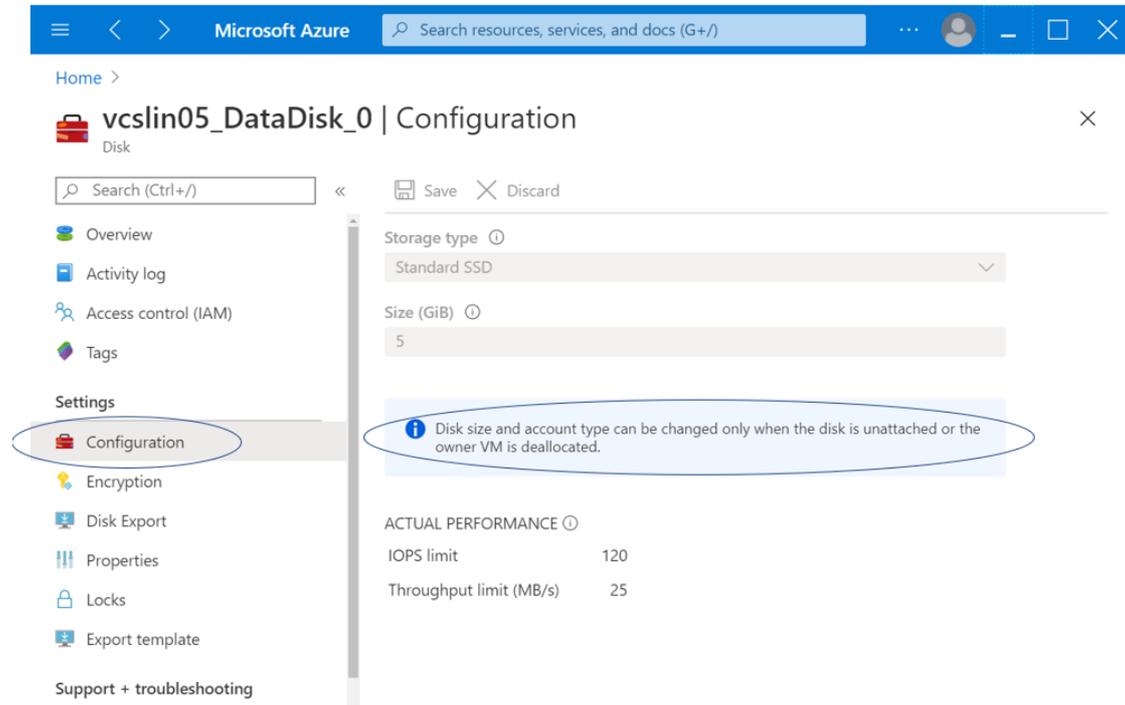
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    5G  0 disk
├─sdc3       8:35  0    5G  0 part
└─sdc8       8:40  0    5G  0 part
```

念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 4.0G 20M 3.8G 1% /data
```

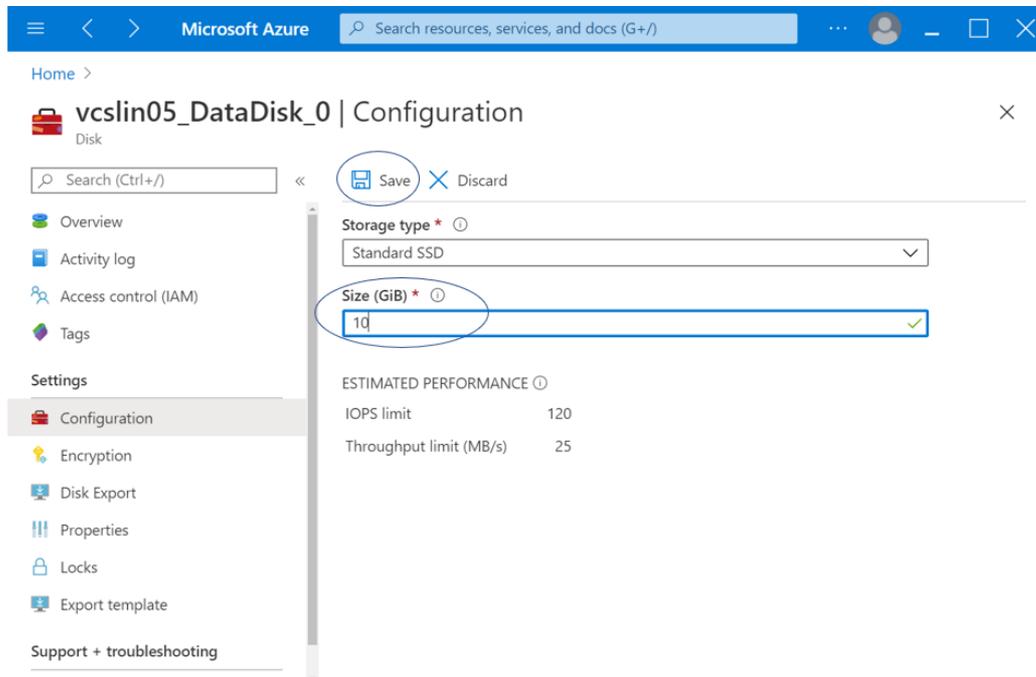
次に、Azure のコンソールで、ブロックストレージの容量を変更します。下記は、拡張対象のブロックストレージを指定し「configuration」を表示した画面です。「Disk size and account type can be changed only when...」の記述の通り、ブロックストレージのサイズ変更は、仮想マシンを停止するか、仮想マシンからブロックストレージを切り離さないといけません。クラスターを停止すると仮想マシンからブロックストレージを切り離せますので、クラスターを停止します。



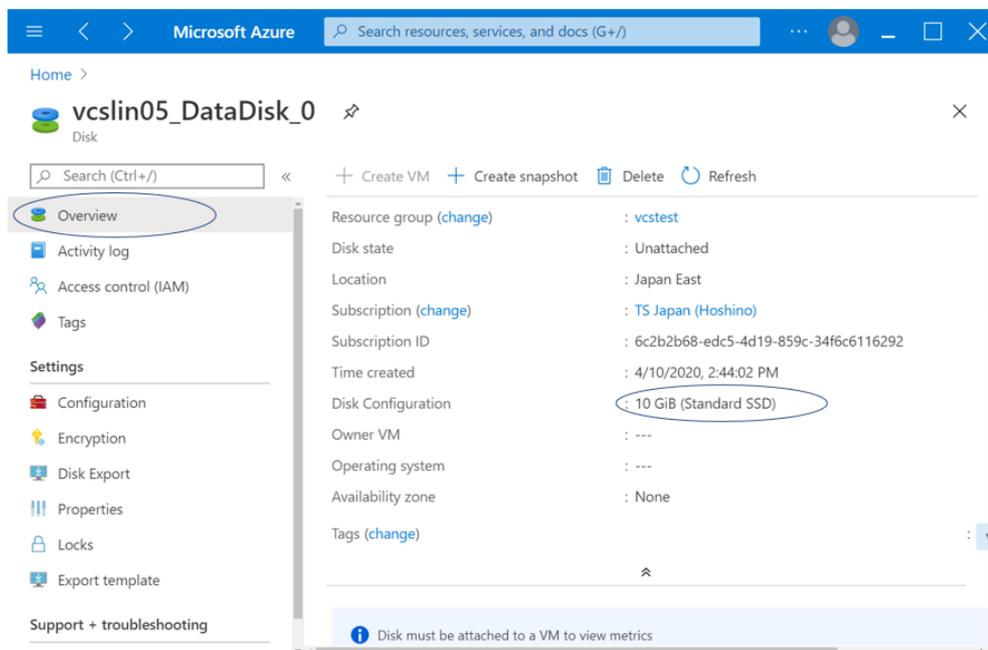
hastop コマンドを使用して、クラスターを停止します。

```
# hastop -all
```

クラスターの停止後、再度 Azure のコンソールで同じ画面を開くと、「Disk size and account type can be changed only when...」の記述が消え、サイズ変更が可能になります。拡張後のサイズを指定し「save」をクリックしてください。



ブロックストレージの「Overview」の画面で、正しくサイズ拡張されたのを確認してください。



ブロックストレージのサイズ拡張が完了したら、hastart コマンドをクラスターの両方のノードで実行し、クラスターを起動します。起動後、hastatus コマンドでどちらのノードが稼働系か確認してください。下記例では、vcslin05 が稼働系です。以降の作業は、全て稼働系で行います。

```
# hastop -all
# hastatus -sum

-- SYSTEM STATE
-- System          State          Frozen
A vcslin05         RUNNING        0
A vcslin06         RUNNING        0

-- GROUP STATE
-- Group           System          Probed         AutoDisabled   State
B application      vcslin05        Y              N               ONLINE
B application      vcslin06        Y              N               OFFLINE
```

まず、lsblk コマンドを用い sdc のサイズを確認します。想定通り 10Gbyte に拡張されています。

```
# lsblk | grep sdc
sdc          8:32  0    10G  0 disk
├─sdc3       8:35  0    10G  0 part
└─sdc8       8:40  0    10G  0 part
```

次に、vxprint コマンドで VxVM が認識しているブロックストレージのサイズを確認します。まだ、10411776 ブロック、すなわち 5Gbyte で認識されています。

```
# vxprint
Disk group: datadg

TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg        datadg         -        -        -        -        -        -
dm 10-193-24-8_disk_0 10-193-24-8_disk_7 10411776 -        -        -
v vol01          fsgen          ENABLED  8388608 -        ACTIVE  -        -
pl vol01-01      vol01          ENABLED  8388608 -        ACTIVE  -        -
sd 10-193-24-8_disk_0-01 vol01-01  ENABLED  8388608 0        -        -        -
```

vxdisk コマンドに、ディスクグループ名である datadg と、再認識対象ディスク名の 10-193-24-8\_disk\_0 を指定して、ディスクサイズを再認識させます。

```
# vxdisk -f -g datadg resize 10-193-24-8_disk_0
```

vxprint で確認すると、確かに 10-193-24-8\_disk\_0 のサイズが 20904976 ブロック、すなわち 10Gbyte と認識されています。

```
# vxprint
Disk group: datadg

TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg        datadg         -        -        -        -        -        -

dm 10-193-24-8_disk_0 10-193-24-8_disk_7 - 20904976 -        -        -

v vol01          fsgen          ENABLED 8388608 -        ACTIVE -        -
pl vol01-01      vol01          ENABLED 8388608 -        ACTIVE -        -
sd 10-193-24-8_disk_0-01 vol01-01  ENABLED 8388608 0        -        -
```

次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 8Gbyte に拡張します。ここで指定するのは、マウントポイント名ではなく、ディスクグループ名（この場合は datadg）とボリューム名（vol01）である事に注意してください。

```
# vxresize -g datadg vol01 8g
```

df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 8.0G 21M 7.5G 1% /data
```

最後に、クラスターの切り替えを行い、切り替え先のノードでサービスグループが正常に Online になり、拡張されたファイルシステムが正常に認識されている事を確認してください。

```
# hagrps -switch application -to vcslin06

# hastatus -sum

-- SYSTEM STATE
-- System          State          Frozen
A vcslin05         RUNNING       0
A vcslin06         RUNNING       0

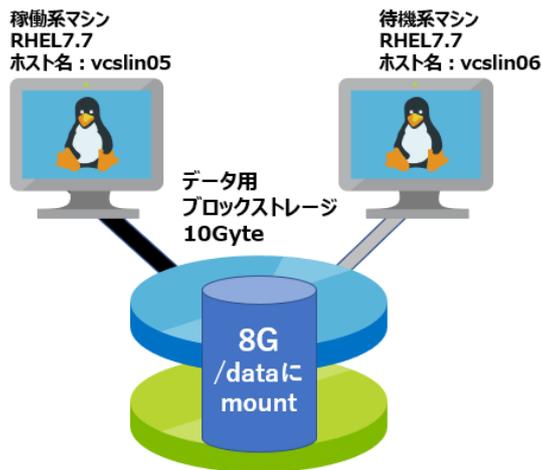
-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B application     vcslin05       Y         N              OFFLINE
B application     vcslin06       Y         N              ONLINE

-- AGENTS FAILED
-- Type           System
J HostMonitor     vcslin05
J HostMonitor     vcslin06

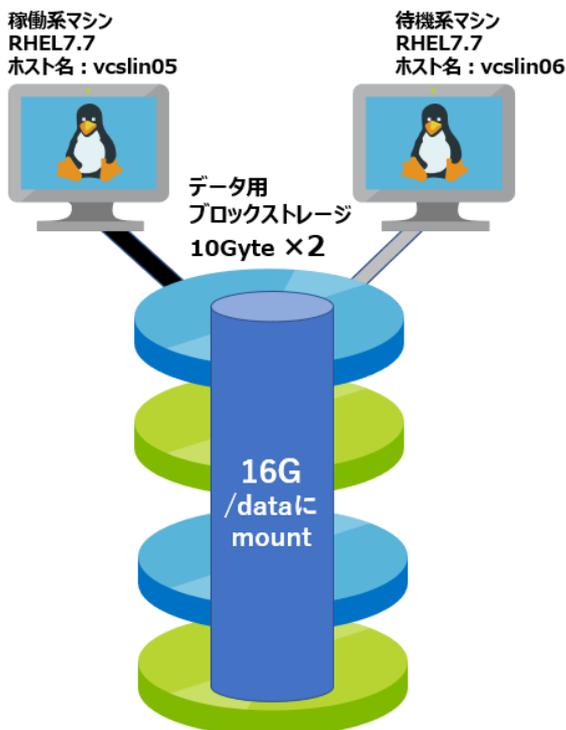
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 8.0G 21M 7.5G 1% /data
```

## 新規にブロックストレージを追加して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 10Gbyte のブロックストレージが共有接続されており（ただし、アクセスできるのは稼働系のみ）、その中に 8Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、稼働系の /data にマウントされます。この状況下で、/data の容量を 16Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。10Gbyte ブロックストレージが追加され、それを使用してデータ用ボリュームとその上位のベリタスファイルシステムが同時に 16Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。拡張前に、InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

まず、VxVM の管理下にあるディスクを確認します。3 本のディスクが確認されていますが、VxVM の管理かである事を示す「STATUS が Online」のディスクは 10-193-24-8\_disk\_7 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
```

次に、10-193-24-8\_disk\_7 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE          ENABLED
sda              10-193-24-8_disk_2 -             -             ENABLED
sdb              10-193-24-8_disk_6 -             -             ENABLED
sdc              10-193-24-8_disk_7 10-193-24-8_disk_0 datadg        ENABLED
```

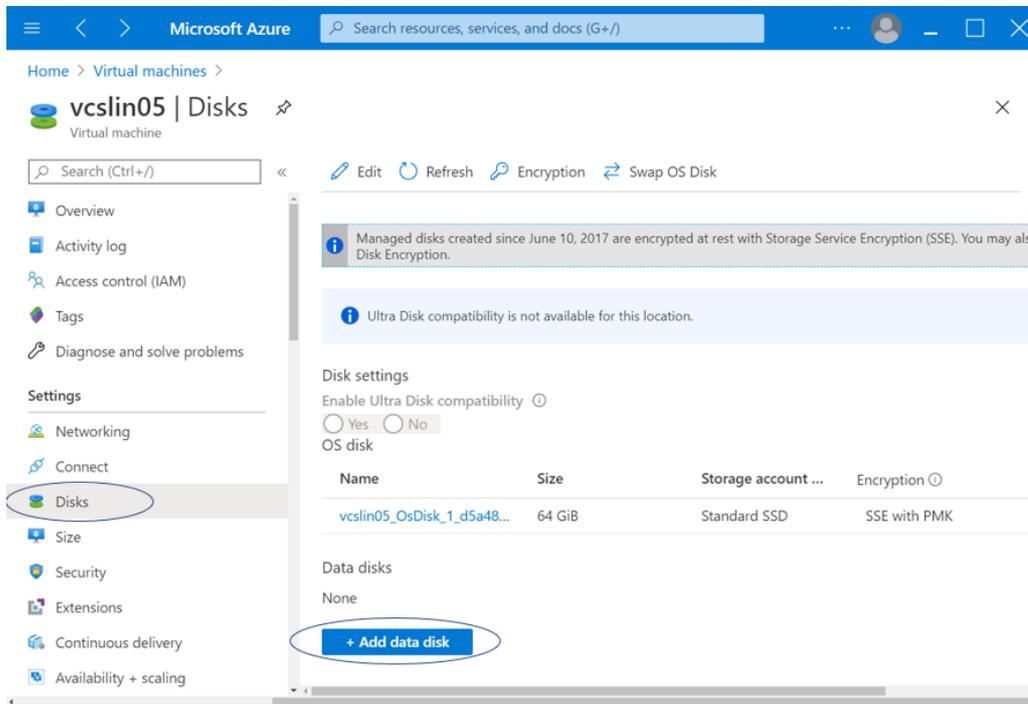
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。10Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    10G  0 disk
├─sdc3       8:35  0    10G  0 part
└─sdc8       8:40  0    10G  0 part
```

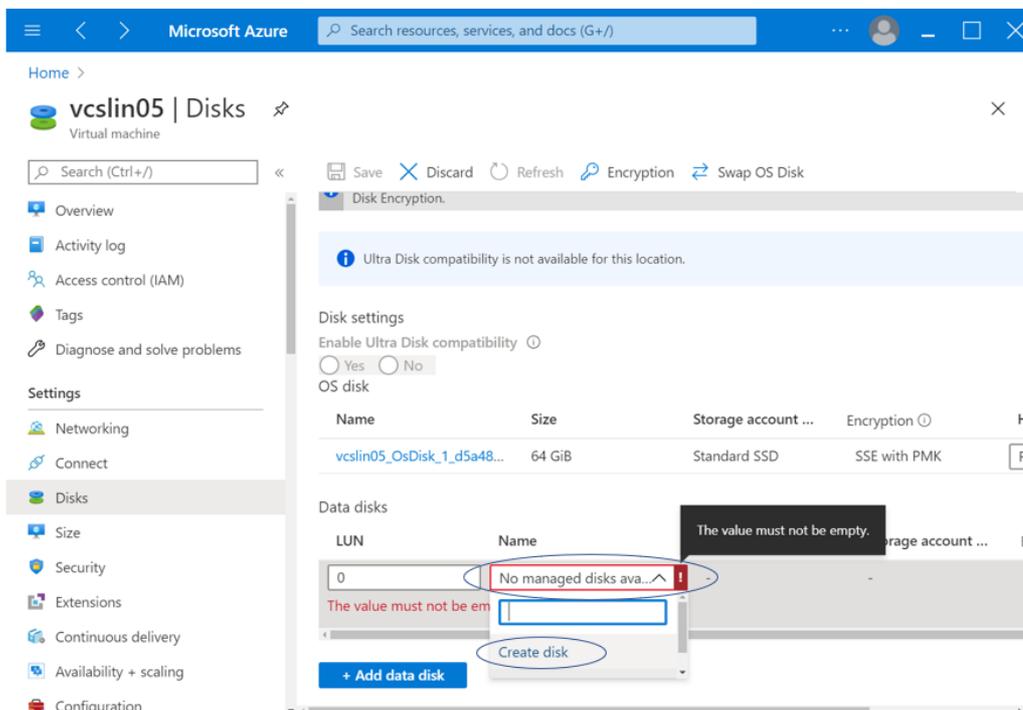
念の為、拡張対象のファイルシステム：/data のサイズを確認します。8Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 8.0G 20M 3.8G 1% /data
```

次に、Azure のコンソールで、ブロックストレージを追加します。ブロックストレージの追加は、仮想マシンを停止しないとできないので、インスタンスを停止してから下記の作業を行ってください。対象の仮想マシンのメニューで「Disk」を指定し、「Add data disk」をクリックしてください。



「Data disks」の欄の「Name」の下のプルダウンメニューを開くと「Create disk」が表示されますので、これをクリックしてください。



「Disk name」と「Resource group」を入力したら「Change size」をクリックしてください。

Home > vcslin05 | Disks >

### Create a managed disk

Create a new disk to store applications and data on your VM. Disk pricing varies based on factors including disk size, storage type, and number of transactions.

**Disk name \***

**Resource group \***

[Create new](#)

**Location**  
Japan East

**Availability zone**  
None

**Source type**  
None

**Size \***  
1024 GiB  
Standard SSD  
[Change size](#)

**Encryption type \***  
(Default) Encryption at-rest with a platform-managed key

[Create](#)

サイズを入力して「OK」をクリックしてください。

Home > v

### Select a disk size

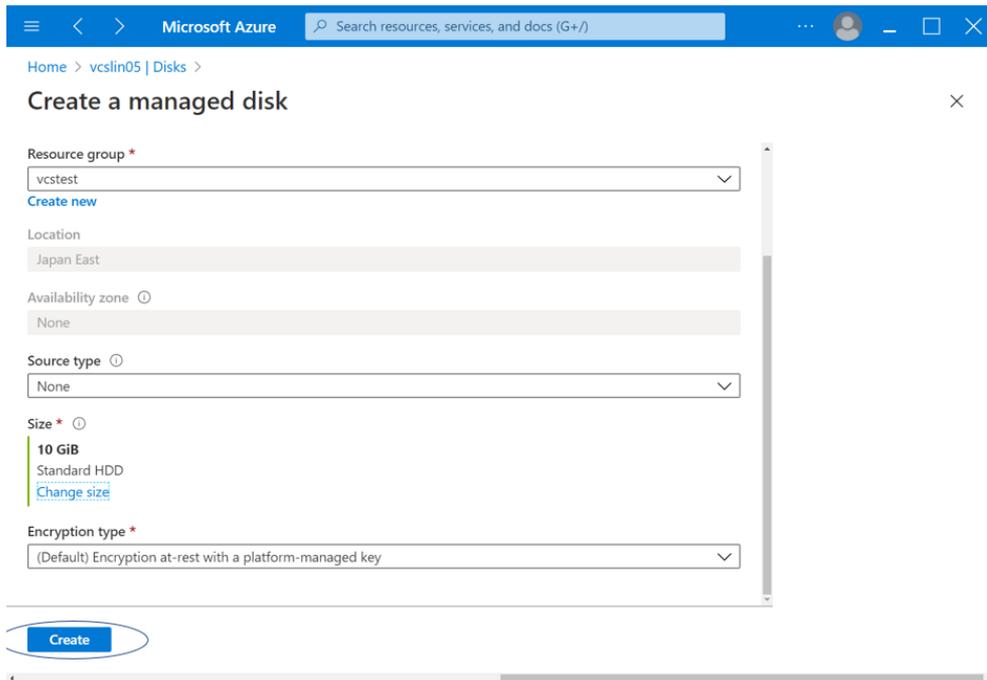
128 GiB	\$10	500	60
256 GiB	\$15	500	60
512 GiB	\$20	500	60
1024 GiB	\$30	500	60
2048 GiB	\$40	500	60
4096 GiB	\$50	500	60
8192 GiB	\$60	1300	300
16384 GiB	\$70	2000	500
32767 GiB	\$80	2000	500

**Create a custom size**  
Enter the size of the disk you would like to create. You will be charged the same rate for your provisioned disk, regardless of how much of the disk space is being used. For example, a 200 GiB disk is provisioned on a 256 GiB disk, so you would be billed for the 256 GiB provisioned.

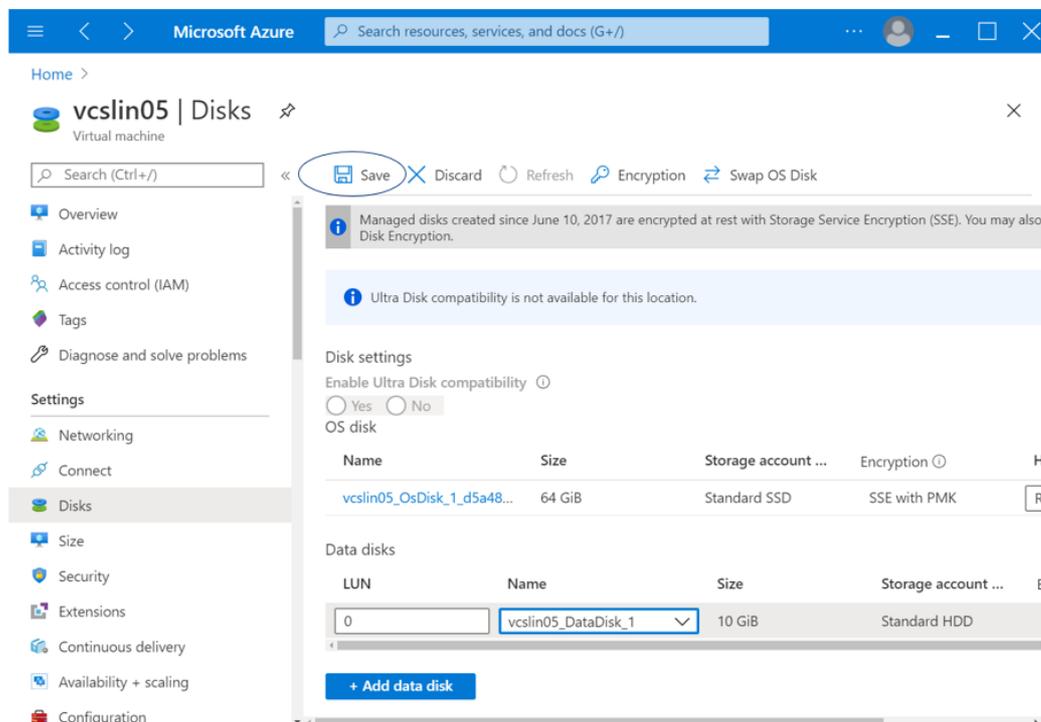
**Custom disk size (GiB) \***

[Create](#) [OK](#)

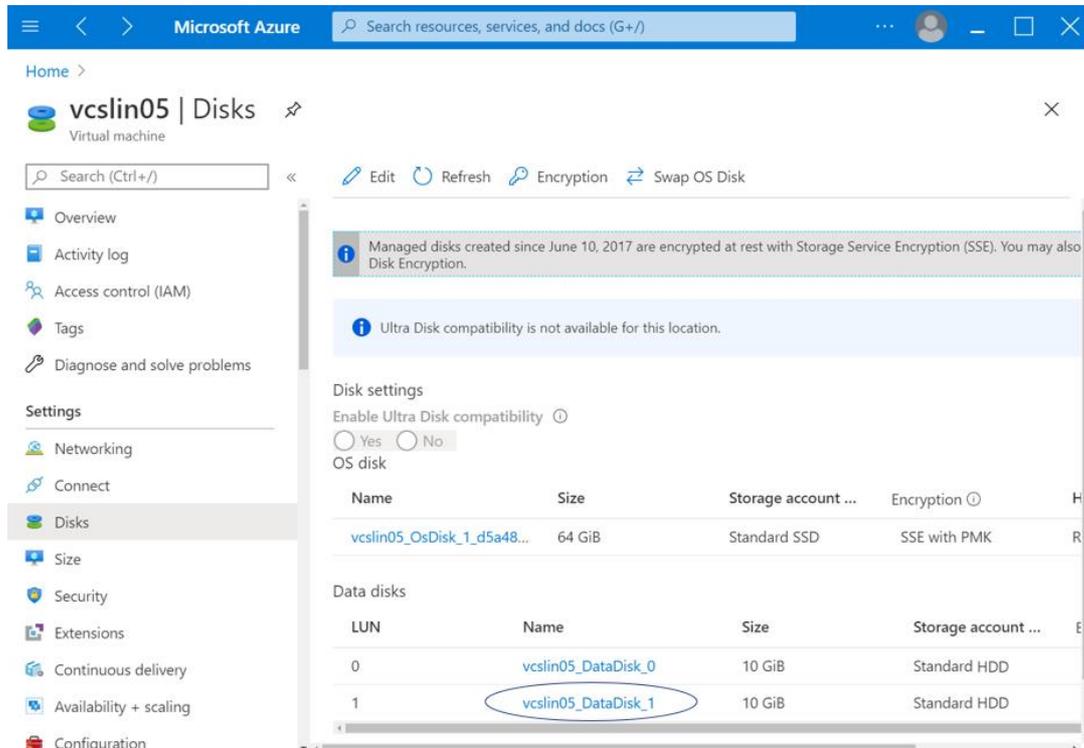
内容を確認して「Create」をクリックしてください。



作成したブロックストレージが「Data disks」欄に表示されますので、Name やサイズなどの内容を確認の上「Save」をクリックしてください。



再度、対象の仮想マシンのメニューで「Disk」を指定すると、作成したブロックストレージが「Data disks」欄に表示されます。確認後、仮想マシンを立ち上げてください。



仮想マシンが立ち上がったら、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda      8:0    0   64G  0 disk
├── sda1  8:1    0   500M  0 part /boot/efi
├── sda2  8:2    0   500M  0 part /boot
├── sda3  8:3    0    2M   0 part
└── sda4  8:4    0    63G  0 part
sdb      8:16   0   50G  0 disk
└── sdb1  8:17   0   50G  0 part /mnt/resource
sdc      8:32   0   10G  0 disk
├── sdc3  8:35   0   10G  0 part
└── sdc8  8:40   0   10G  0 part
sdd      8:48   0   10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前で認識されているか確認します。10-193-24-8\_disk\_8 という名前で認識されている事がわかりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              10-193-24-8_disk_2  -              -              ENABLED
sdb              10-193-24-8_disk_6  -              -              ENABLED
sdc              10-193-24-8_disk_7  10-193-24-8_disk_0  datadg        ENABLED
sdd              10-193-24-8_disk_8  -              -              ENABLED
```

vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:none     -             -             online invalid
```

次に、vxdisksetup コマンドで 10-193-24-8\_disk\_8 をイニシャライズします。

```
# vxdisksetup -i 10-193-24-8_disk_8
```

再度 vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 の STATUS は online に変わり、VxVM の管理下になった事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:cdsdisk  -             -             online
```

vxchg コマンドで、既存のディスクグループ：datadg に 10-193-24-8\_disk\_8 を加えます。vxdisk list コマンドで確認すると、10-193-24-8\_disk\_8 が datadg に所属した事が確認できます。これで、追加されたディスク：10-193-24-8\_disk\_8 を用いて既存のボリューム：vol01 を拡張する準備が整いました。

```
# vxchg -g datadg adddisk 10-193-24-8_disk_8
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
10-193-24-8_disk_2 auto:LVM      -             -             LVM
10-193-24-8_disk_6 auto:none     -             -             online invalid
10-193-24-8_disk_7 auto:cdsdisk  10-193-24-8_disk_0 datadg        online
10-193-24-8_disk_8 auto:cdsdisk  10-193-24-8_disk_8 datadg        online
```

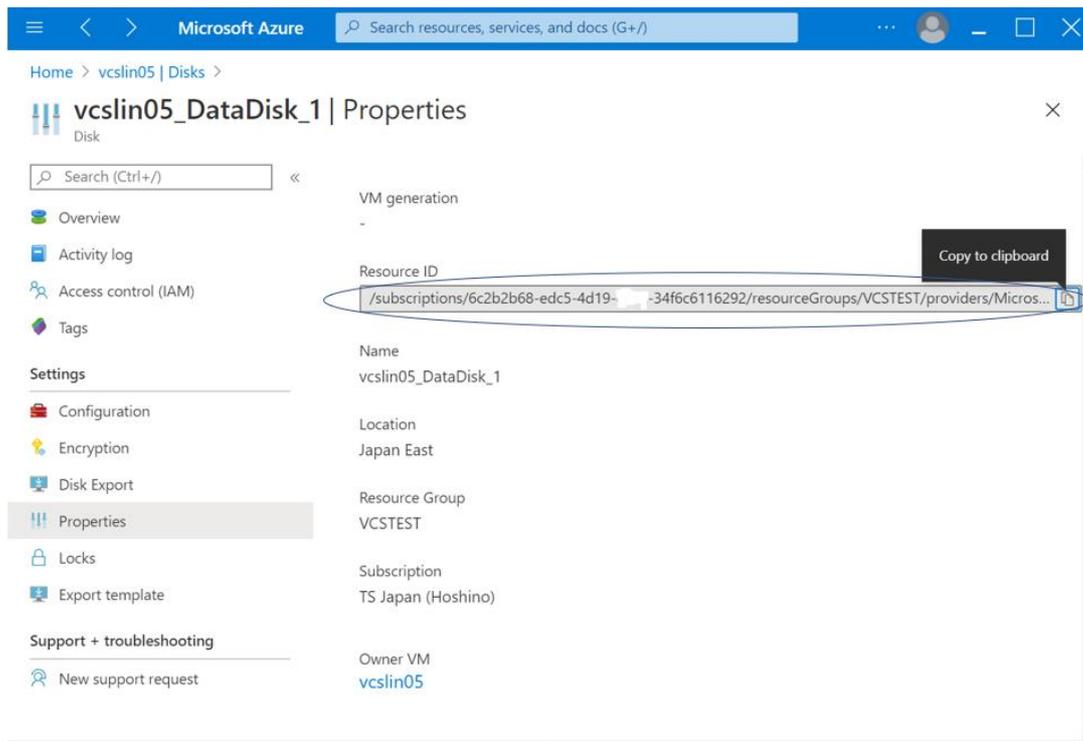
次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 16Gbyte に拡張します。ここで指定するのは、マウントポイント名ではなく、ディスクグループ名（この場合は datadg）とボリューム名（vol01）である事に注意してください。

```
# vxresize -g datadg vol01 16g
```

df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 16G 23M 15G 1% /data
```

クラスター構成に新規のブロックストレージを追加した場合は、そのブロックストレージをクラスターの管理下に置く必要があります。そのためには、ブロックストレージの Resource ID が必要です。Azure のコンソールで、Resource ID を確認してください。下記例では、Resource ID は「/subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/VCSTEST/providers/Microsoft.Compute/disks/vcslin05\_DataDisk\_1」です。これを、クラスターの AzureDisk リソースの DiskIds パラメーターに追加する必要があります。AzureDisk リソースの DiskIds パラメーターに関する詳細は、[https://www.veritas.com/support/en\\_US/doc/InfoScale\\_7.4.1\\_RHEL\\_on\\_Azure\\_deploy\\_PrivatelP\\_の32ページ付近](https://www.veritas.com/support/en_US/doc/InfoScale_7.4.1_RHEL_on_Azure_deploy_PrivatelP_の32ページ付近)を参照してください。



クラスターの構成を変更するにあたり、初めにクラスターを変更可能モードにします。

```
# haconf -makerw
```

次に、hares コマンドの -display オプションを用いて、既に AzureDisk リソースの DiskIds パラメーターに登録されている既存ディスクの Resource ID を確認します。下記例では、既存ディスクの Resource ID は「/subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/vctest/providers/Microsoft.Compute/disks/vcslin05\_DataDisk\_0」です。

```
# hares -display azuredisk | grep DiskIds
. . . (中略) . . .
azuredisk DiskIds global /subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/vctest/providers/Microsoft.Compute/disks/vcslin05_DataDisk_0
```

次に、hares コマンドの -modify オプションで、azuredisk リソースの DiskIds パラメータを更新します。パラメータとして、既存ディスクの ResoudeID と新しく追加したディスクの ResourceID をスペースで区切って記述します。

```
# hares -modify azuredisk DiskIds /subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/vctest/providers/Microsoft.Compute/disks/vcslin05_DataDisk_0 /subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/VCSTEST/providers/Microsoft.Compute/disks/vcslin05_DataDisk_1
```

再度、hares コマンドの -display オプションを用いて、既に AzureDisk リソースの DiskIds パラメータに計 2 つのディスクの ResoudeID が追加された事を確認します。

```
# hares -display azuredisk | grep DiskIds
. . . (中略) . . .
azuredisk DiskIds global /subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/vctest/providers/Microsoft.Compute/disks/vcslin05_DataDisk_0 /subscriptions/6c2b2b68-edc5-4d19-xxxx-34f6c6116292/resourceGroups/VCSTEST/providers/Microsoft.Compute/disks/vcslin05_DataDisk_1
```

新規のブロックストレージが正しくクラスタの管理下に置かれた事を確認するために、クラスタの切り替えを行い、切り替え先のノードでサービスグループが正常に Online になり、拡張されたファイルシステムが正常に認識されている事を確認してください。

```
# hagr -switch application -to vcslin06
# hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A vcslin05         RUNNING        0
A vcslin06         RUNNING        0
-- GROUP STATE
-- Group           System         Probed         AutoDisabled   State
B application      vcslin05       Y              N               OFFLINE
B application      vcslin06       Y              N               ONLINE
-- AGENTS FAILED
-- Type           System
J HostMonitor     vcslin05
J HostMonitor     vcslin06
# df -h | grep /data
/dev/vx/dsk/datadg/vol01 16G 23M 15G 1% /data
```

最後に、クラスタを変更不可モードにして、作業終了です。

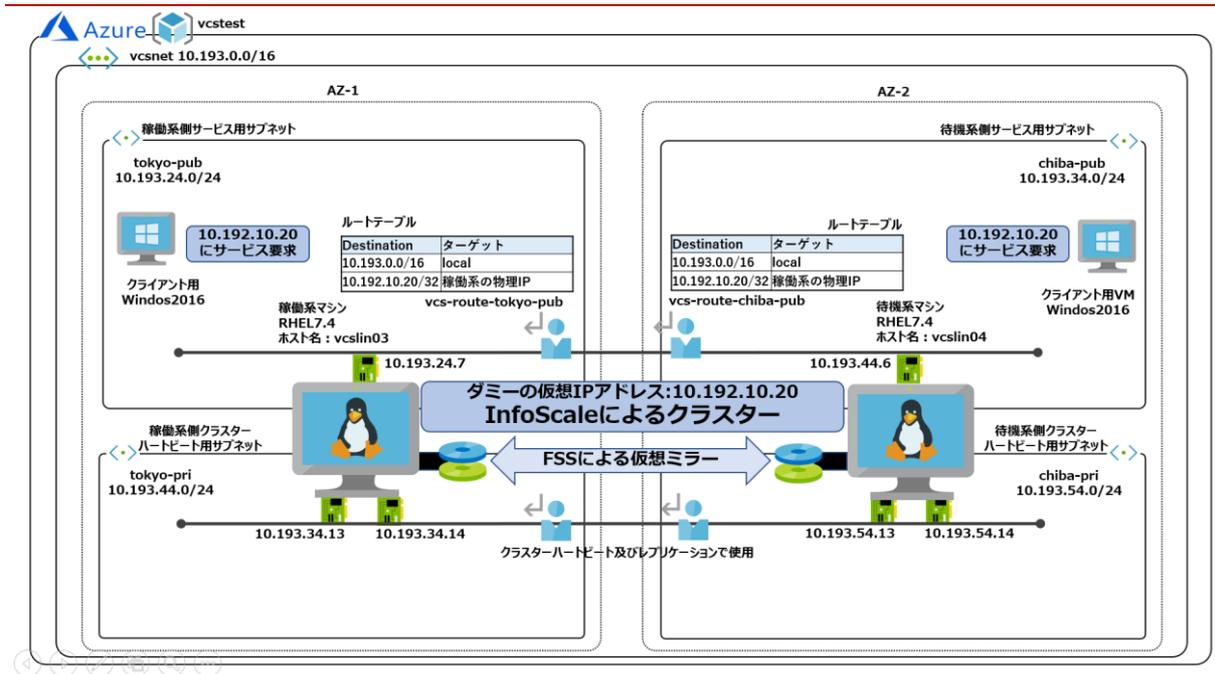
```
# haconf -dump -makero
```

## 4. FSS を用いた仮想ミラー型クラスター構成でのデータ領域拡張手順

この章では、InfoScale の FSS 機能による仮想ミラー型クラスター上でデータ領域を拡張するシナリオを説明します。尚、Azure 上で InfoScale の FSS 機能による仮想ミラー型クラスターを構築する手順については、ベリタスより詳細が公開されています。詳しくは

[https://www.veritas.com/support/en\\_US/doc/InfoScale\\_7.4.1\\_RHEL\\_on\\_Azure\\_deploy\\_FSS\\_OverlayIP](https://www.veritas.com/support/en_US/doc/InfoScale_7.4.1_RHEL_on_Azure_deploy_FSS_OverlayIP) を参照してください。

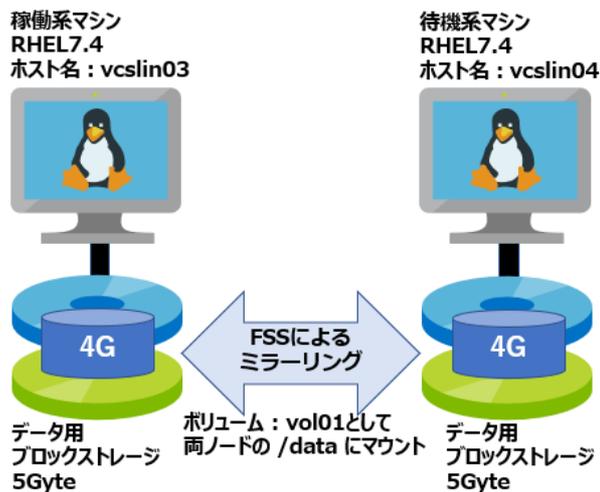
### 本章の前提となるシステム構成



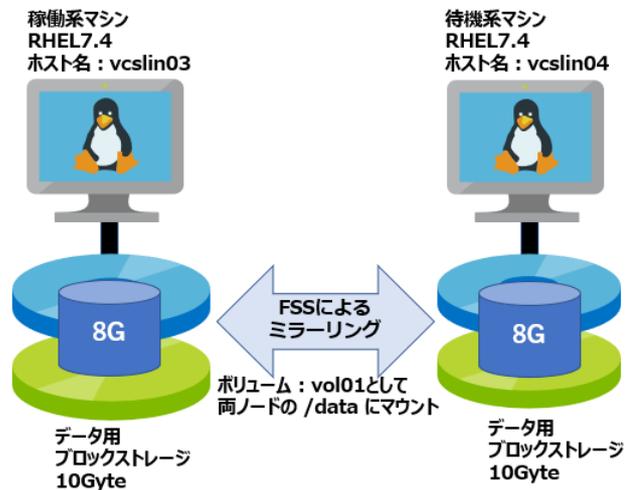
前提となる構成イメージ

## 既存のブロックストレージを拡張して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 5Gbyte のブロックストレージがそれぞれ接続されており、それをミラーリングした 4Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、両ノードの /data にクラスターマウントされます。この状況下で、/data の容量を 8Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。ブロックストレージが 10Gbyte に拡張され、データ用ボリュームとその上位のベリタスファイルシステムが同時に 8Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。まず、拡張前に InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

1号機で VxVM の管理下にあるディスクを確認します。4本のディスクが確認されていますが、10-193-34-6\_disk\_5 は、FSS 機能によって見えている2号機のディスクなので、1号機で VxVM の管理下である事を示す「STATUS が online exported shared」のディスクは 10-193-24-7\_disk\_0 の1つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none     -             -             online invalid
10-193-24-7_disk_4 auto:none     -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk  10-193-24-7_disk_0 fssdg         online exported shared
10-193-34-6_disk_5 auto:cdsdisk  10-193-34-6_disk_0 fssdg         online shared remote
```

次に、10-193-24-7\_disk\_0 が OS からどのような名前で認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda             sda             -             -             ENABLED
sdb             10-193-24-7_disk_4 -             -             ENABLED
sdc             10-193-24-7_disk_6 10-193-24-7_disk_0 fssdg         ENABLED
```

さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc             8:32  0      5G  0 disk
├─sdc3          8:35  0      5G  0 part
└─sdc8          8:40  0      5G  0 part
```

念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 4.0G 38M 3.5G 1% /data
```

同じ確認作業を2号機でも行います。2号機で VxVM の管理下にあるディスクを確認します。4本のディスクが確認されていますが、10-193-24-7\_disk\_0 は、FSS 機能によって見えている1号機のディスクなので、2号機で VxVM の管理下である事を示す「STATUS が online exported shared」のディスクは 10-193-34-6\_disk\_0 の1つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none     -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk  10-193-24-7_disk_0 fssdg         online shared remote
10-193-34-6_disk_5 auto:cdsdisk  10-193-34-6_disk_0 fssdg         online exported shared
10-193-34-6_disk_6 auto:none     -             -             online invalid
```

次に、10-193-34-6\_disk\_0 が OS からどのような名前で認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
```

SUBPATH	DANAME	DMNAME	GROUP	STATE
sda	sda	-	-	ENABLED
sdc	10-193-34-6_disk_5	10-193-34-6_disk_0	fssdg	ENABLED
sdb	10-193-34-6_disk_6	-	-	ENABLED

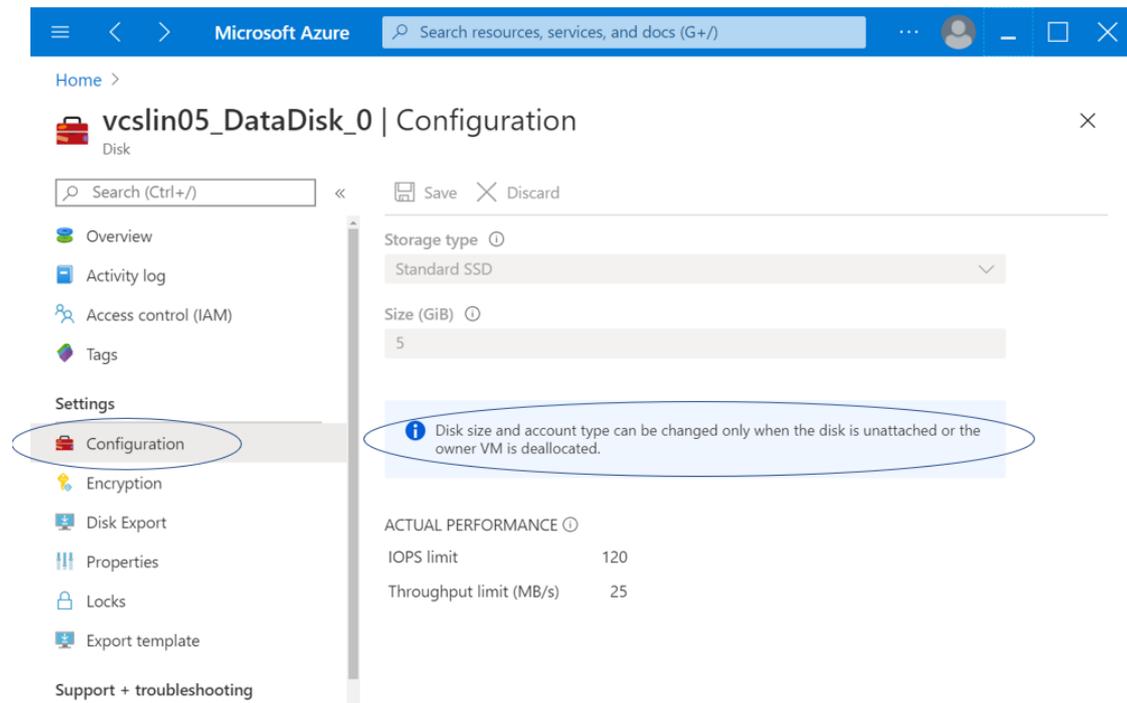
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc      8:32  0    5G  0 disk
├─sdc3   8:35  0    5G  0 part
└─sdc8   8:40  0    5G  0 part
```

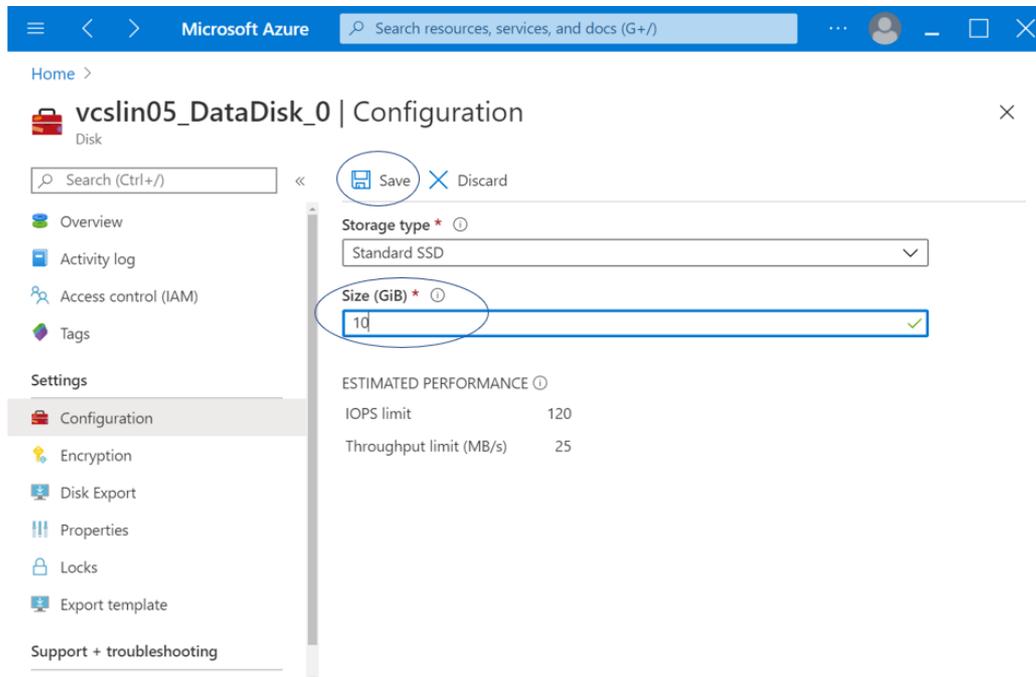
念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 4.0G  38M  3.5G  1% /data
```

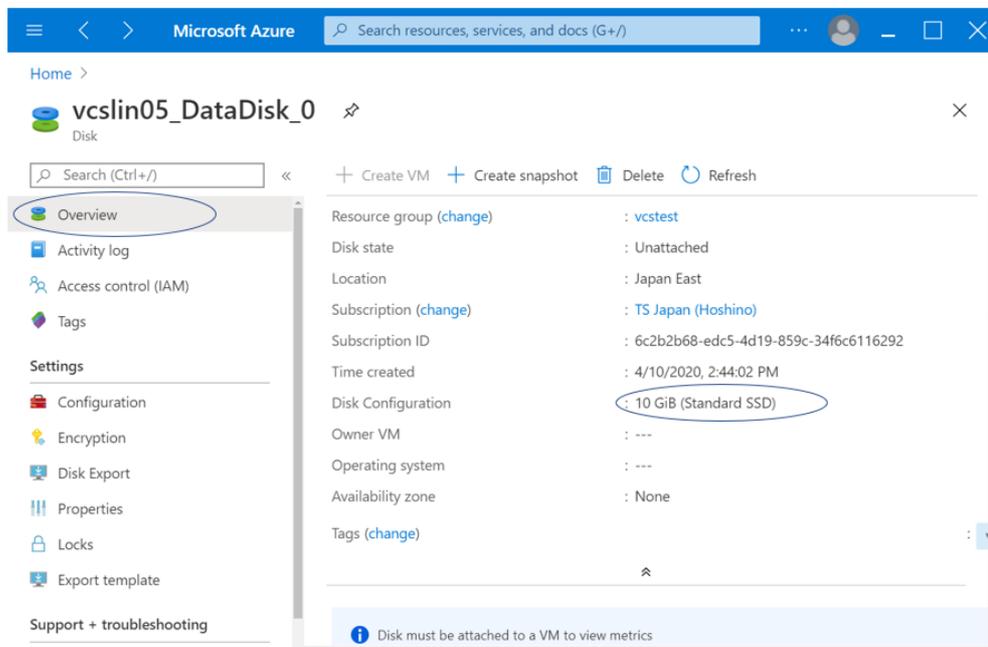
次に、Azure のコンソールで、ブロックストレージの容量を変更します。下記は、拡張対象のブロックストレージを指定し「configuration」を表示した画面です。「Disk size and account type can be changed only when...」の記述の通り、ブロックストレージのサイズ変更は、仮想マシンを停止するか、仮想マシンからブロックストレージを切り離さないとできません。FSS 構成の場合、クラスターを停止しても仮想マシンからブロックストレージを切り離すことはできませんので、仮想マシンを停止します。



クラスター構成内の全ての仮想マシンの停止後、再度 Azure のコンソールで同じ画面を開くと、「Disk size and account type can be changed only when...」の記述が消え、サイズ変更が可能になります。拡張後のサイズを指定し「save」をクリックしてください。



ブロックストレージの「Overview」の画面で、正しくサイズ拡張されたのを確認してください。尚、ここまでのブロックストレージ追加の作業はクラスター内の全ノードに対して行ってください。



ブロックストレージのサイズ拡張が完了したら、仮想マシンを起動し、正常にクラスターが立ち上がった事を確認してください。クラスターが正常に立ち上がったら、1号機と2号機でlsblkコマンドを用いsdcのサイズを確認します。想定通り10Gbyteに拡張されています。

```
# lsblk | grep sdc
sdc          8:32  0   10G  0 disk
├── sdc3     8:35  0   10G  0 part
└── sdc8     8:40  0   10G  0 part
```

次に、vxprintコマンドでVxVMが認識しているブロックストレージのサイズを確認します。10-193-24-7\_disk\_0も10-193-34-6\_disk\_0もサイズは10411776ブロック、すなわち5Gbyteで認識されています。

```
# vxprint
Disk group: fssdg

TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg fssdg         fssdg         -        -        -        -        -        -
dm 10-193-24-7_disk_0 10-193-24-7_disk_6 10411776 -        -        -
dm 10-193-34-6_disk_0 10-193-34-6_disk_5 10411776 REMOTE  -        -
:
(以下略)
:
```

新しいデータ用ブロックストレージのサイズをInfoScaleに認識させるためにはコマンドの実行が必要ですが、再認識させるコマンドは、Cluster Volume Manager（以下CVMと記述）のマスターノードからのみ実行可能です。まず、vxclustadmコマンドを使用してCVMのマスターノードを確認します。下記の通り、1号機（vcslin03）がマスターである事が確認できました。

```
# vxclustadm nidmap
Name          CVM Nid  CM Nid  State
vcslin03     1         0      Joined: Master
vcslin04     0         1      Joined: Slave
```

1号機で、vxdiskコマンドに、ディスクグループ名であるfssdgと、再認識対象ディスク名の10-193-24-7\_disk\_0を指定して、ディスクサイズを再認識させます。

```
# vxdisk -g fssdg resize 10-193-24-7_disk_0
```

vxprint で確認すると、確かに 10-193-24-7\_disk\_0 のサイズが 20974976 ブロック、すなわち 10Gbyte と認識されています。

```
# vxprint
Disk group: fssdg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg fssdg      fssdg      -        -        -        -        -        -
dm 10-193-24-7_disk_0 10-193-24-7_disk_6 - 20904976 - - - -
dm 10-193-34-6_disk_0 10-193-34-6_disk_5 - 10411776 - REMOTE - -
v vol01      fsgen      ENABLED 8388608 -        ACTIVE - -
pl vol01-01   vol01      ENABLED 8388608 -        ACTIVE - -
sd 10-193-24-7_disk_0-01 vol01-01 ENABLED 6291456 0 - - -
sd 10-193-24-7_disk_0-03 vol01-01 ENABLED 2097152 6291456 - - -
pl vol01-02   vol01      ENABLED 8388608 -        ACTIVE - -
sd 10-193-34-6_disk_0-01 vol01-02 ENABLED 6291456 0 - - -
sd 10-193-34-6_disk_0-03 vol01-02 ENABLED 2097152 6291456 - - -
dc vol01_dco  vol01      -        -        -        -        - -
v vol01_dcl  gen        ENABLED 67840 -        ACTIVE - -
pl vol01_dcl-01 vol01_dcl  ENABLED 67840 -        ACTIVE - -
sd 10-193-24-7_disk_0-02 vol01_dcl-01 ENABLED 67840 0 - - -
pl vol01_dcl-02 vol01_dcl  ENABLED 67840 -        ACTIVE - -
sd 10-193-34-6_disk_0-02 vol01_dcl-02 ENABLED 67840 0 - - -
```

次に、CVM のマスターを 2 号機 (vcslin04) に切り替えます。この作業は 1 号機にて行ってください。

```
# vxclustadm setmaster vcslin04
# vxclustadm nidmap
Name          CVM Nid  CM Nid  State
vcslin03      1         0      Joined: Slave
vcslin04      0         1      Joined: Master
```

2 号機にログインし、vxdisk コマンドに、ディスクグループ名である fssdg と、再認識対象ディスク名の 10-193-34-6\_disk\_0 を指定して、ディスクサイズを再認識させます。

```
# vxdisk -g fssdg resize 10-193-34-6_disk_0
```

vxprint で確認すると、10-193-34-6\_disk\_0 のサイズも 20974976 ブロック、すなわち 10Gbyte と認識されています。

```
# vxprint
Disk group: fssdg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg fssdg      fssdg      -        -        -        -        -        -
dm 10-193-24-7_disk_0 10-193-24-7_disk_6 - 20904976 - REMOTE - -
dm 10-193-34-6_disk_0 10-193-34-6_disk_5 - 20904976 - - - -
v vol01      fsgen      ENABLED 8388608 -        ACTIVE - -
pl vol01-01   vol01      ENABLED 8388608 -        ACTIVE - -
sd 10-193-24-7_disk_0-01 vol01-01 ENABLED 6291456 0 - - -
sd 10-193-24-7_disk_0-03 vol01-01 ENABLED 2097152 6291456 - - -
pl vol01-02   vol01      ENABLED 8388608 -        ACTIVE - -
:
(以下略)
:
```

CVM のマスターは、切り替わったままでも通常の運用には影響ありませんので、切り戻す必要はありません。ただし、この後の作業は、CVM のマスターである 2 号機で行ってください。

次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 8Gbyte に拡張します。

```
# vxresize -g fssdg vol01 8g
```

vxprint コマンドを実行すると、vol01 のサイズが 16777216 ブロック、すなわち 8Gbyte に拡張されたことが確認できます。

```
# vxprint
Disk group: fssdg
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg	fssdg	fssdg	-	-	-	-	-	-
dm	10-193-24-7_disk_0	10-193-24-7_disk_6	-	20904976	-	REMOTE	-	-
dm	10-193-34-6_disk_0	10-193-34-6_disk_5	-	20904976	-	-	-	-
v	vol01	fsgen	ENABLED	16777216	-	ACTIVE	-	-
pl	vol01-01	vol01	ENABLED	16777216	-	ACTIVE	-	-
sd	10-193-24-7_disk_0-01	vol01-01	ENABLED	6291456	0	-	-	-
sd	10-193-24-7_disk_0-03	vol01-01	ENABLED	10485760	6291456	-	-	-
pl	vol01-02	vol01	ENABLED	16777216	-	ACTIVE	-	-
sd	10-193-34-6_disk_0-01	vol01-02	ENABLED	6291456	0	-	-	-
sd	10-193-34-6_disk_0-03	vol01-02	ENABLED	10485760	6291456	-	-	-
dc	vol01_dco	vol01	-	-	-	-	-	-
v	vol01_dcl	gen	ENABLED	67840	-	ACTIVE	-	-
pl	vol01_dcl-01	vol01_dcl	ENABLED	67840	-	ACTIVE	-	-
sd	10-193-24-7_disk_0-02	vol01_dcl-01	ENABLED	67840	0	-	-	-
pl	vol01_dcl-02	vol01_dcl	ENABLED	67840	-	ACTIVE	-	-
sd	10-193-34-6_disk_0-02	vol01_dcl-02	ENABLED	67840	0	-	-	-

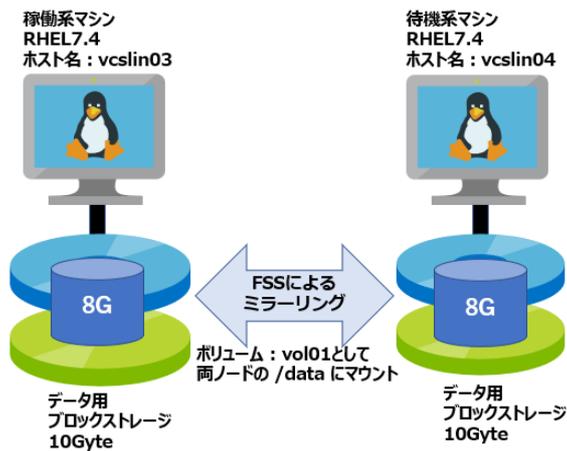
df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 8.0G 38M 7.5G 1% /data
```

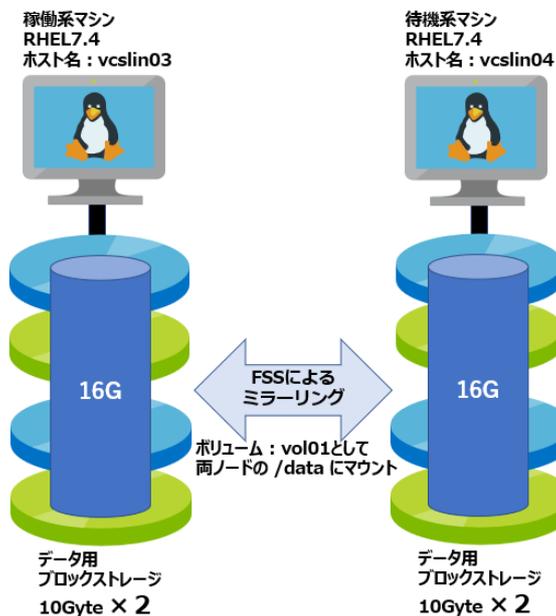
これで、作業終了です。

## 新規にブロックストレージを追加して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 10Gbyte のブロックストレージがそれぞれ接続されており、それをミラーリングした 8Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、両ノードの /data にクラスターマウントされます。この状況下で、/data の容量を 16Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。10Gbyte ブロックストレージが追加され、それを使用してデータ用ボリュームとその上位のベリタスファイルシステムが同時に 16Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。拡張前に、InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

1 号機で VxVM の管理下にあるディスクを確認します。4 本のディスクが確認されていますが、10-193-34-6\_disk\_5 は、FSS 機能によって見えている 2 号機のディスクなので、1 号機で VxVM の管理下である事を示す「STATUS が online exported shared」のディスクは 10-193-24-7\_disk\_0 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda              auto:none     -             -             online invalid
10-193-24-7_disk_4 auto:none     -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk  10-193-24-7_disk_0 fssdg         online exported shared
10-193-34-6_disk_5 auto:cdsdisk  10-193-34-6_disk_0 fssdg         online shared remote
```

次に、10-193-24-7\_disk\_0 が OS からどのような名前で見られているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -             -             ENABLED
sdb              10-193-24-7_disk_4 -             -             ENABLED
sdc              10-193-24-7_disk_6 10-193-24-7_disk_0 fssdg         ENABLED
```

さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc              8:32  0      5G  0 disk
├─sdc3           8:35  0      5G  0 part
└─sdc8           8:40  0      5G  0 part
```

念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 4.0G  38M  3.5G  1% /data
```

同じ確認作業を 2 号機でも行います。2 号機で VxVM の管理下にあるディスクを確認します。4 本のディスクが確認されていますが、10-193-24-7\_disk\_0 は、FSS 機能によって見えている 1 号機のディスクなので、2 号機で VxVM の管理下である事を示す「STATUS が online exported shared」のディスクは 10-193-34-6\_disk\_0 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda              auto:none     -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk  10-193-24-7_disk_0 fssdg         online shared remote
10-193-34-6_disk_5 auto:cdsdisk  10-193-34-6_disk_0 fssdg         online exported shared
10-193-34-6_disk_6 auto:none     -             -             online invalid
```

次に、10-193-34-6\_disk\_0 が OS からどのような名前で認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -              -              ENABLED
sdc              10-193-34-6_disk_5  10-193-34-6_disk_0  fssdg          ENABLED
sdb              10-193-34-6_disk_6  -              -              ENABLED
```

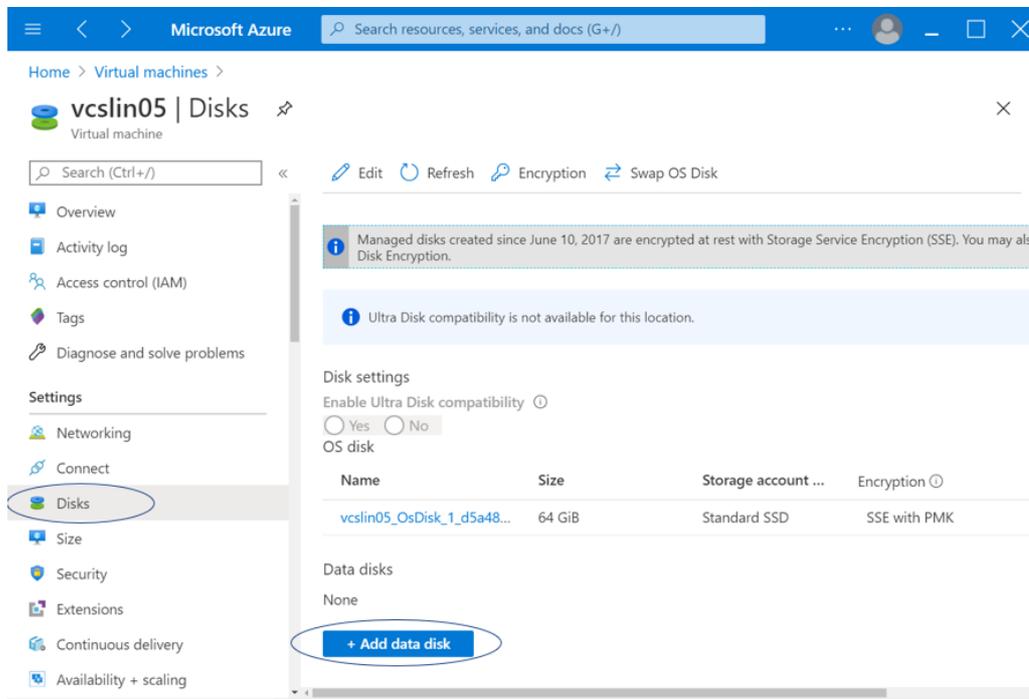
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    5G  0 disk
├─sdc3       8:35  0    5G  0 part
└─sdc8       8:40  0    5G  0 part
```

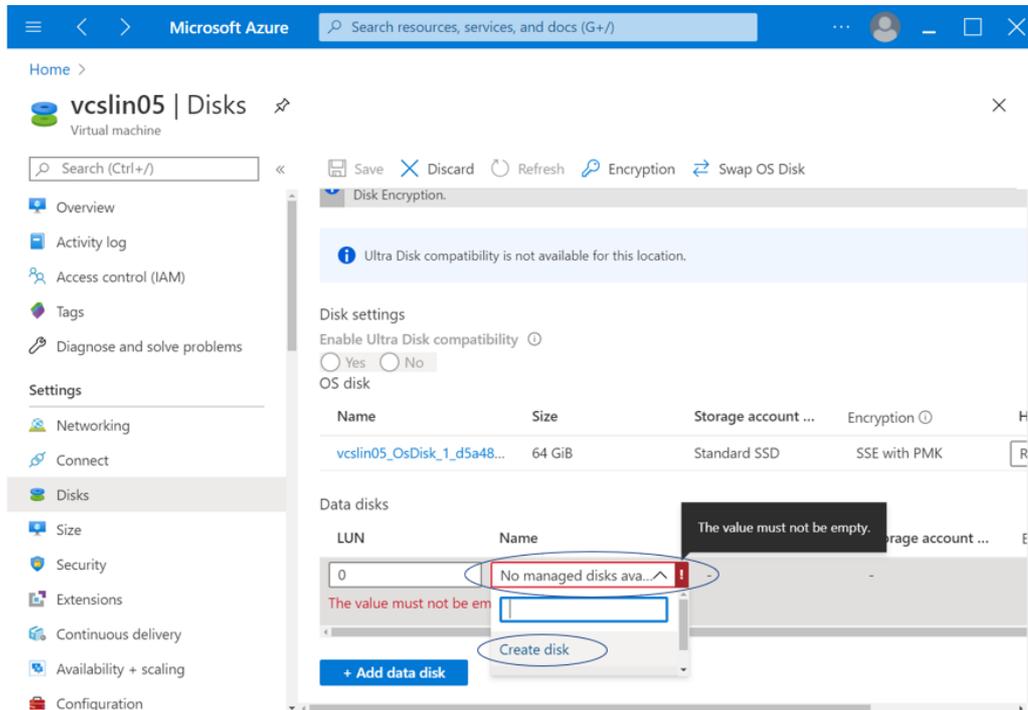
念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 4.0G  38M  3.5G  1% /data
```

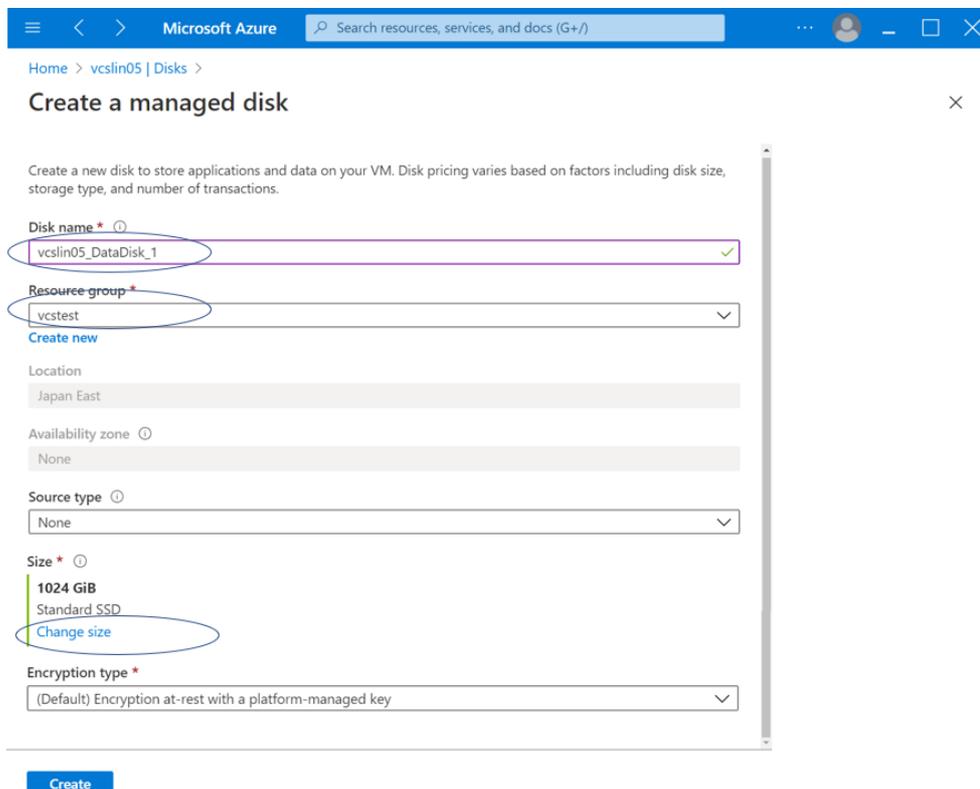
次に、Azure のコンソールで、ブロックストレージを追加します。ブロックストレージの追加は、仮想マシンを停止しないとできないので、インスタンスを停止してから下記の作業を行ってください。対象の仮想マシンのメニューで「Disk」を指定し、「Add data disk」をクリックしてください。



「Data disks」の欄の「Name」の下のプルダウンメニューを開くと「Create disk」が表示されますので、これをクリックしてください。



「Disk name」と「Resource group」を入力したら「Change size」をクリックしてください。



サイズを入力して「OK」をクリックしてください。

The screenshot shows the 'Select a disk size' dialog in the Microsoft Azure portal. The dialog has a title bar with 'Microsoft Azure' and a search bar. Below the title bar, there is a breadcrumb 'Home > v...' and a close button 'X'. The main content area contains a table of disk sizes and prices, and a custom size input field.

Size	Price	Price	Price
128 GiB	\$10	500	60
256 GiB	\$15	500	60
512 GiB	\$20	500	60
1024 GiB	\$30	500	60
2048 GiB	\$40	500	60
4096 GiB	\$50	500	60
8192 GiB	\$60	1300	300
16384 GiB	\$70	2000	500
32767 GiB	\$80	2000	500

Below the table, there is a section for 'Create a custom size'. It includes a text input field for 'Custom disk size (GiB) \*' with the value '10' and a green checkmark. Below the input field, there are two buttons: 'Create' and 'OK'.

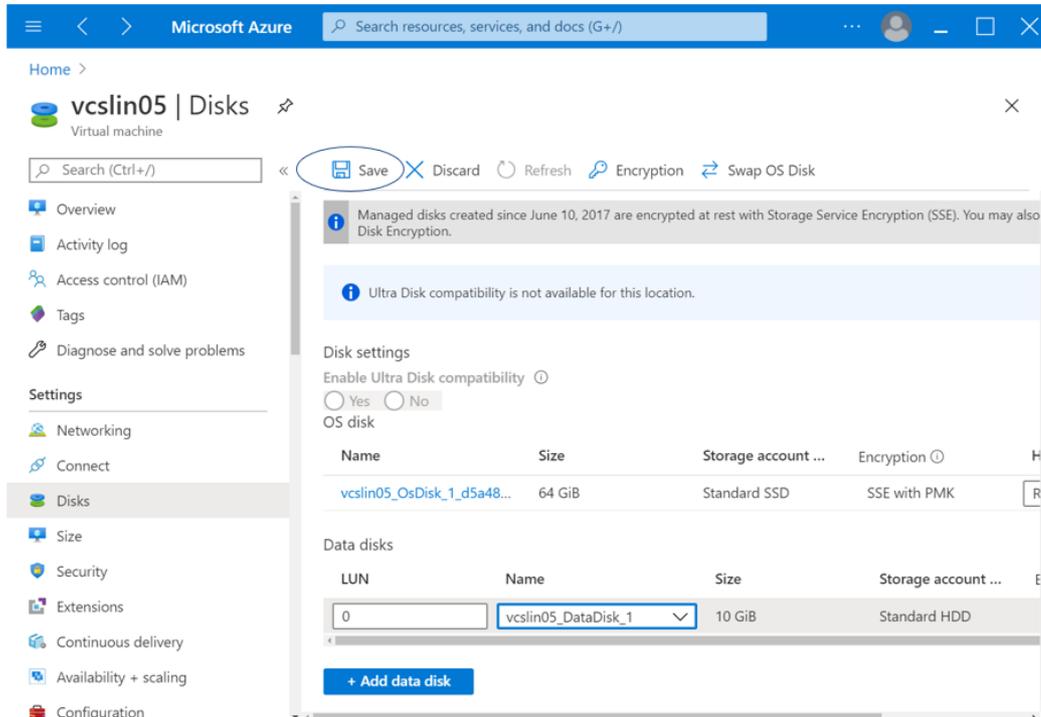
内容を確認して「Create」をクリックしてください。

The screenshot shows the 'Create a managed disk' dialog in the Microsoft Azure portal. The dialog has a title bar with 'Microsoft Azure' and a search bar. Below the title bar, there is a breadcrumb 'Home > vmlin05 | Disks >' and a close button 'X'. The main content area contains several configuration options:

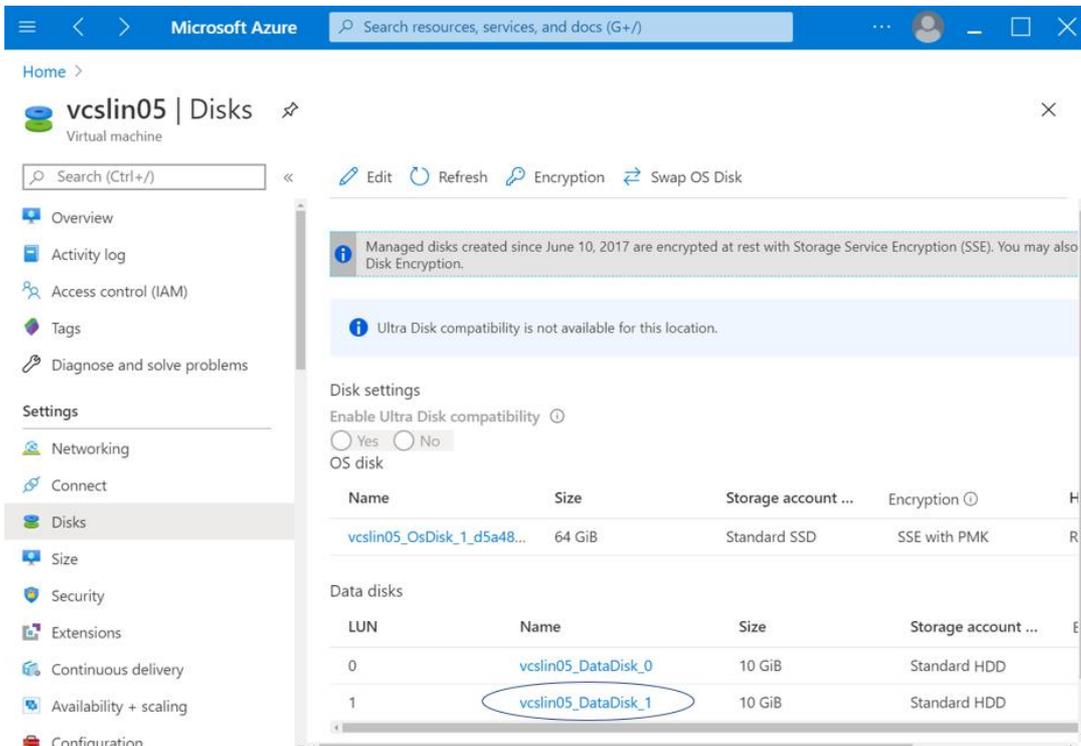
- Resource group \***: A dropdown menu with 'vcstest' selected.
- Location**: A dropdown menu with 'Japan East' selected.
- Availability zone**: A dropdown menu with 'None' selected.
- Source type**: A dropdown menu with 'None' selected.
- Size \***: A dropdown menu with '10 GiB' selected. Below it, there is a link 'Change size'.
- Encryption type \***: A dropdown menu with '(Default) Encryption at-rest with a platform-managed key' selected.

At the bottom of the dialog, there is a blue button labeled 'Create' circled in blue.

作成したブロックストレージが「Data disks」欄に表示されますので、Name やサイズなどの内容を確認の上「Save」をクリックしてください。



再度、対象の仮想マシンのメニューで「Disk」を指定すると、作成したブロックストレージが「Data disks」欄に表示されます。確認後、仮想マシンを立ち上げてください。尚、ここまでのブロックストレージ追加の作業はクラスター内の全ノードに対して行ってください。



1号機が立ち上がったら、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda                8:0    0   64G  0 disk
├── sda1            8:1    0   500M  0 part /boot/efi
├── sda2            8:2    0   500M  0 part /boot
├── sda3            8:3    0     2M  0 part
└── sda4            8:4    0    63G  0 part
sdb                8:16   0    50G  0 disk
└── sdb1            8:17   0    50G  0 part /mnt/resource
sdc                8:32   0    10G  0 disk
├── sdc3            8:35   0    10G  0 part
└── sdc8            8:40   0    10G  0 part
sdd                8:48   0    10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前でも認識されているか確認します。10-193-24-7\_disk\_5 という名前で認識されている事がわかりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -              -              ENABLED
sdb              10-193-24-7_disk_4 -              -              ENABLED
sdd              10-193-24-7_disk_5 -              -              ENABLED
sdc              10-193-24-7_disk_6 10-193-24-7_disk_0 fssdg         ENABLED
```

vxdisk list コマンドで確認すると、10-193-24-7\_disk\_5 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-7_disk_4 auto:none    -             -             online invalid
10-193-24-7_disk_5 auto:none    -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk 10-193-24-7_disk_0 fssdg         online exported shared
10-193-34-6_disk_5 auto:cdsdisk 10-193-34-6_disk_0 fssdg         online shared remote
```

次に、vxdisksetup コマンドで 10-193-24-7\_disk\_5 をイニシャライズし、さらに export します。

```
# vxdisksetup -i 10-193-24-7_disk_5
# vxdisk export 10-193-24-7_disk_5
```

再度 vxdisk list コマンドで確認すると、10-193-24-7\_disk\_5 の STATUS は online exported に変わり、VxVM の管理下になり且つ 2号機に共有された事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-7_disk_4 auto:none    -             -             online invalid
10-193-24-7_disk_5 auto:cdsdisk -             -             online exported
10-193-24-7_disk_6 auto:cdsdisk 10-193-24-7_disk_0 fssdg         online exported shared
10-193-34-6_disk_5 auto:cdsdisk 10-193-34-6_disk_0 fssdg         online shared remote
```

同じ作業を 2 号機でも行います。まず、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda                8:0    0   64G  0 disk
├── sda1            8:1    0   500M  0 part /boot/efi
├── sda2            8:2    0   500M  0 part /boot
├── sda3            8:3    0     2M  0 part
└── sda4            8:4    0    63G  0 part
sdb                8:16   0    50G  0 disk
└── sdb1            8:17   0    50G  0 part /mnt/resource
sdc                8:32   0    10G  0 disk
├── sdc3            8:35   0    10G  0 part
└── sdc8            8:40   0    10G  0 part
sdd                8:48   0    10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前で認識されているか確認します。10-193-34-6\_disk\_4 という名前で認識されている事がわかりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -              -              ENABLED
sdd              10-193-34-6_disk_4 -              -              ENABLED
sdc              10-193-34-6_disk_5 10-193-34-6_disk_0 fssdg         ENABLED
sdb              10-193-34-6_disk_6 -              -              ENABLED
```

vxdisk list コマンドで確認すると、10-193-34-6\_disk\_4 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk 10-193-24-7_disk_0 fssdg         online shared remote
10-193-24-7_disk_5 auto:cdsdisk -             -             online shared remote
10-193-34-6_disk_4 auto:none    -             -             online invalid
10-193-34-6_disk_5 auto:cdsdisk 10-193-34-6_disk_0 fssdg         online exported shared
10-193-34-6_disk_6 auto:none    -             -             online invalid
```

次に、vxdisksetup コマンドで 10-193-34-6\_disk\_4 をイニシャライズし、さらに export します。

```
# vxdisksetup -i 10-193-34-6_disk_4
# vxdisk export 10-193-34-6_disk_4
```

再度 vxdisk list コマンドで確認すると、10-193-34-6\_disk\_4 の STATUS は online exported に変わり、VxVM の管理下になり且つ 1 号機に共有された事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk 10-193-24-7_disk_0 fssdg         online shared remote
10-193-24-7_disk_5 auto:cdsdisk -             -             online shared remote
10-193-34-6_disk_4 auto:cdsdisk -             -             online exported shared
10-193-34-6_disk_5 auto:cdsdisk 10-193-34-6_disk_0 fssdg         online exported shared
10-193-34-6_disk_6 auto:none    -             -             online invalid
```

vxvg コマンドで、既存のディスクグループ：fssdg に、先に追加した 2 つのディスクである 10-193-24-7\_disk\_5 と 10-193-34-6\_disk\_4 を追加します。この作業は、どちらか一方のノードで行います。

```
# vxvg -g fssdg adddisk 10-193-24-7_disk_5 10-193-34-6_disk_4
```

vxdisk list で確認すると、10-193-24-7\_disk\_5 と 10-193-34-6\_disk\_4 が fssdg に所属している事が確認できます。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda              auto:none     -             -             online invalid
10-193-24-7_disk_6 auto:cdsdisk  10-193-24-7_disk_0 fssdg         online shared remote
10-193-24-7_disk_5 auto:cdsdisk  10-193-24-7_disk_1 fssdg         online shared remote
10-193-34-6_disk_4 auto:cdsdisk  10-193-34-6_disk_1 fssdg         online exported shared
10-193-34-6_disk_5 auto:cdsdisk  10-193-34-6_disk_0 fssdg         online exported shared
10-193-34-6_disk_6 auto:none     -             -             online invalid
```

次に、vxresize コマンドを使用して、ボリュームとファイルシステムのサイズを同時に 16Gbyte に拡張します。

```
# vxresize -g fssdg vol01 16g
```

df コマンドで拡張対象のファイルシステム：/data のサイズが 16Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 16.0G  38M  3.5G  1% /data
```

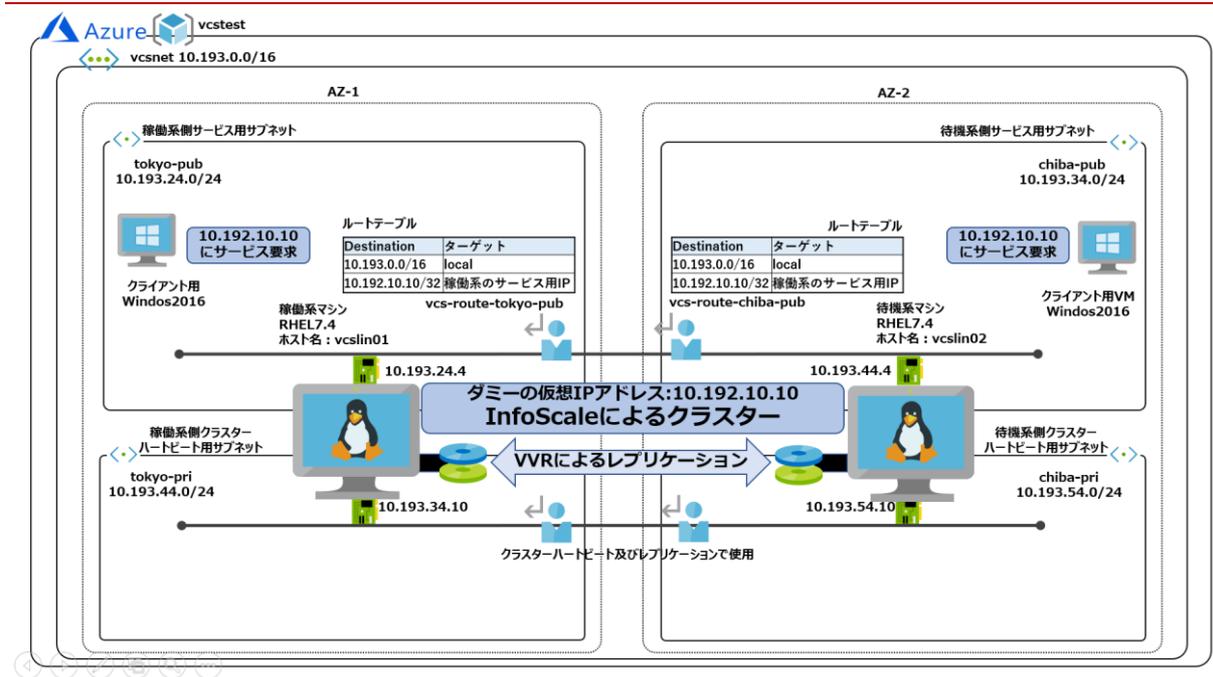
これで作業は終了です。

## 5. VVR を用いたレプリケーション型クラスター構成でのデータ領域拡張手順

この章では、InfoScale の VVR 機能によるレプリケーション型クラスター上でデータ領域を拡張するシナリオを説明します。尚、Azure 上で InfoScale の VVR 機能によるレプリケーション型クラスターを構築する手順については、ベリタスより詳細が公開されています。詳しくは

[https://www.veritas.com/support/en\\_US/doc/InfoScale\\_7.4.1\\_RHEL\\_on\\_Azure\\_deploy\\_VVR\\_Overlay\\_IP](https://www.veritas.com/support/en_US/doc/InfoScale_7.4.1_RHEL_on_Azure_deploy_VVR_Overlay_IP) を参照してください。

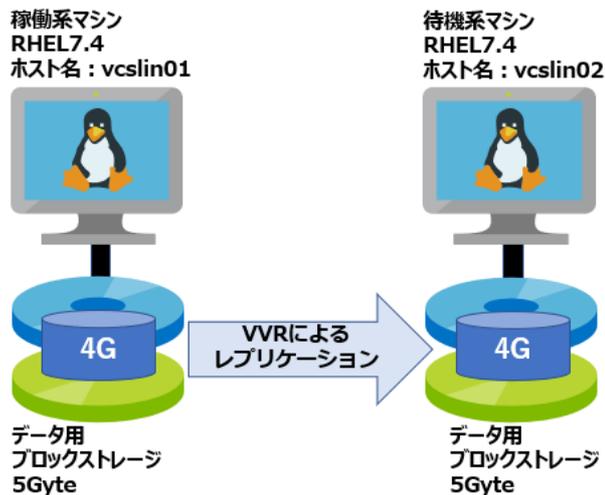
### 本章の前提となるシステム構成



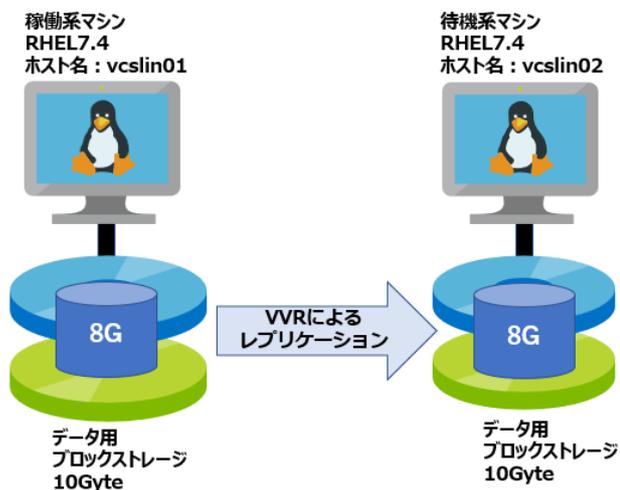
前提となる構成イメージ

## 既存のブロックストレージを拡張して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 5Gbyte のブロックストレージがそれぞれ接続されており、それをレプリケーションした 4Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、稼働系マシンの/data にマウントされます。この状況下で、/data の容量を 8Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。ブロックストレージが 10Gbyte に拡張され、データ用ボリュームとその上位のベリタスファイルシステムが同時に 8Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。まず、拡張前に InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

1 号機で VxVM の管理下にあるディスクを確認します。VxVM の管理下であることを示す「STATUS が online」のディスクは 10-193-24-4\_disk\_1 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-4_disk_4 auto:cdsdisk 10-193-24-4_disk_1 datadg         online
10-193-24-4_disk_7 auto:none    -             -             online invalid
```

次に、10-193-24-4\_disk\_1 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -             -             ENABLED
sdc              10-193-24-4_disk_4 10-193-24-4_disk_1 datadg         ENABLED
sdb              10-193-24-4_disk_7 -             -             ENABLED
```

さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    5G  0 disk
├─sdc3       8:35  0    5G  0 part
└─sdc8       8:40  0    5G  0 part
```

念の為、拡張対象のファイルシステム：/data のサイズを確認します。4Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/datavol 4.0G  20M  3.8G  1% /data
```

同じ確認作業を 2 号機でも行います。2 号機で VxVM の管理下にあるディスクを確認します。VxVM の管理下であることを示す「STATUS が online」のディスクは 10-193-34-4\_disk\_1 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-34-4_disk_6 auto:none    -             -             online invalid
10-193-34-4_disk_7 auto:cdsdisk 10-193-34-4_disk_1 datadg         online
```

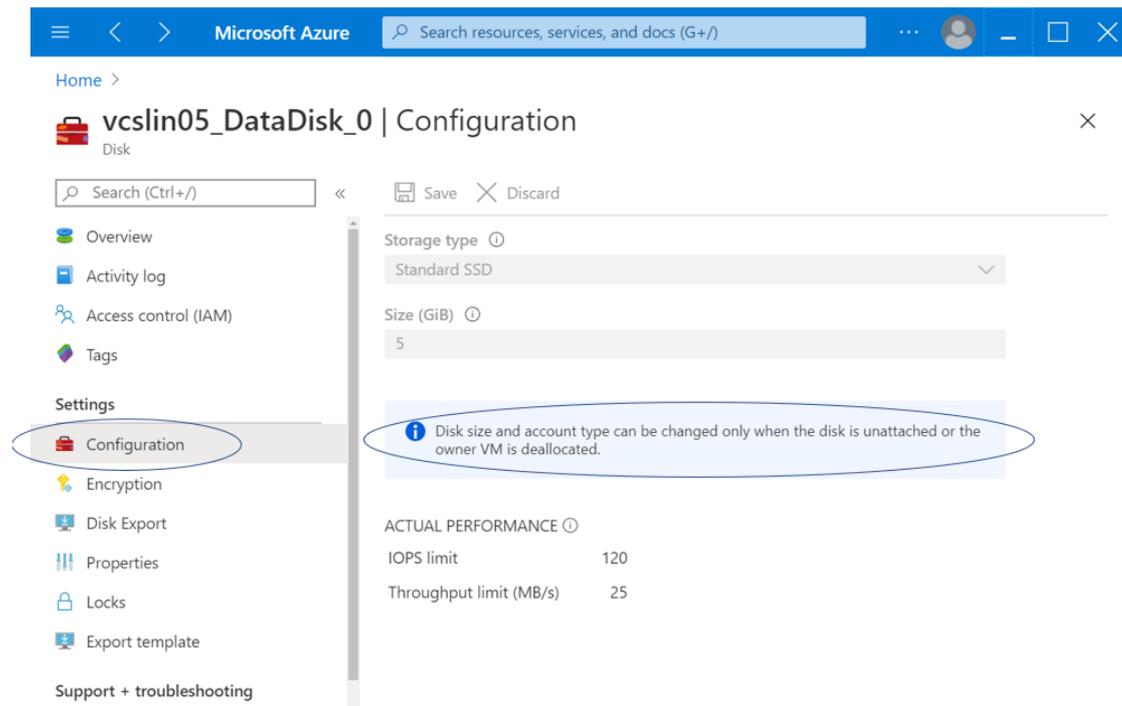
次に、10-193-34-4\_disk\_1 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -             -             ENABLED
sdb              10-193-34-4_disk_6 -             -             ENABLED
sdc              10-193-34-4_disk_7 10-193-34-4_disk_1 datadg         ENABLED
```

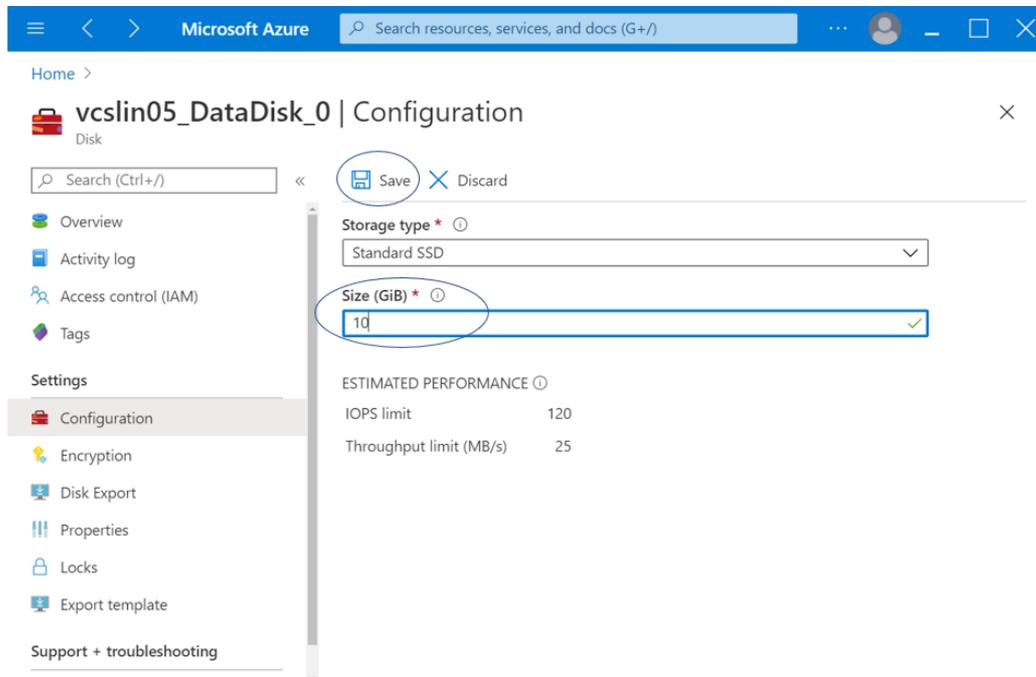
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。5Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    5G  0 disk
├── sdc3     8:35  0    5G  0 part
└── sdc8     8:40  0    5G  0 part
```

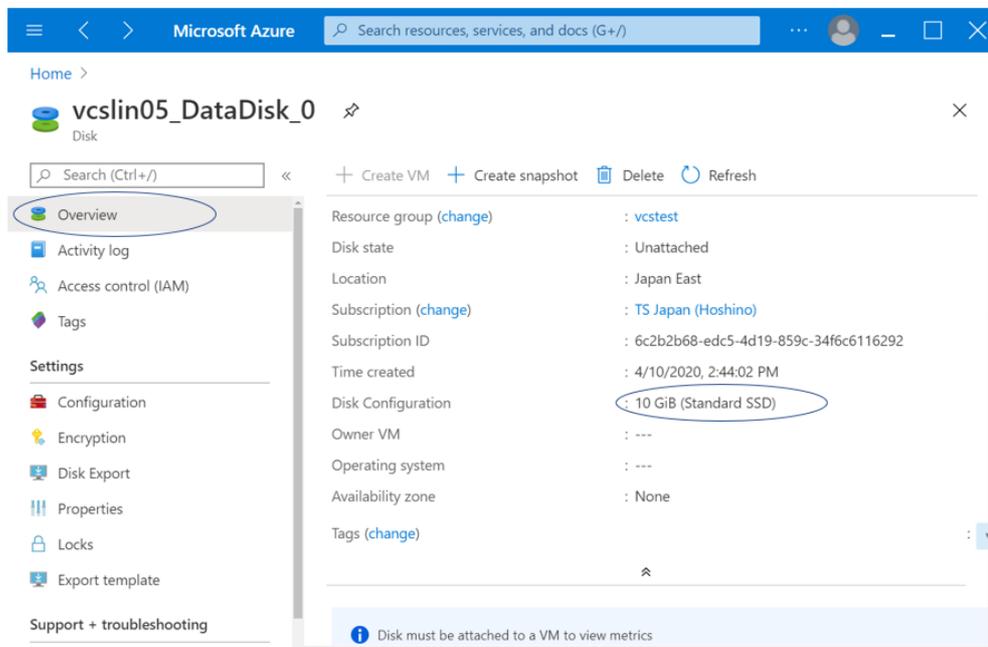
次に、Azure のコンソールで、ブロックストレージの容量を変更します。下記は、拡張対象のブロックストレージを指定し「configuration」を表示した画面です。「Disk size and account type can be changed only when...」の記述の通り、ブロックストレージのサイズ変更は、仮想マシンを停止するか、仮想マシンからブロックストレージを切り離さないといけません。VVR 構成の場合、クラスターを停止しても仮想マシンからブロックストレージを切り離すことはできませんので、仮想マシンを停止します。



クラスター構成内の全ての仮想マシンの停止後、再度 Azure のコンソールで同じ画面を開くと、「Disk size and account type can be changed only when...」の記述が消え、サイズ変更が可能になります。拡張後のサイズを指定し「save」をクリックしてください。



ブロックストレージの「Overview」の画面で、正しくサイズ拡張されたのを確認してください。尚、ここまでのブロックストレージ追加の作業はクラスター内の全ノードに対して行ってください。



ブロックストレージのサイズ拡張が完了したら、仮想マシンを起動し、正常にクラスターが立ち上がった事を確認してください。クラスターが正常に立ち上がったら、1号機と2号機でlsblkコマンドを用いsdcのサイズを確認します。想定通り10Gbyteに拡張されています。

```
# lsblk | grep sdc
sdc          8:32  0   10G  0 disk
├── sdc3      8:35  0   10G  0 part
└── sdc8      8:40  0   10G  0 part
```

1号機で、vxprintコマンドでVxVMが認識しているブロックストレージのサイズを確認します。10-193-24-4\_disk\_1のサイズは10411776ブロック、すなわち5Gbyteで認識されています。

```
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -
dm 10-193-24-4_disk_1 10-193-24-4_disk_10 - 10411776 - - -
:
(以下略)
:
```

同じように2号機でもvxprintコマンドでVxVMが認識しているブロックストレージのサイズを確認します。10-193-34-4\_disk\_1のサイズは10411776ブロック、すなわち5Gbyteで認識されています。

```
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -
dm 10-193-34-4_disk_1 10-193-34-4_disk_7 - 10411776 - - -
:
(以下略)
:
```

1号機で、vxdiskコマンドに、ディスクグループ名であるdatadgと、再認識対象ディスク名の10-193-24-4\_disk\_1を指定して、ディスクサイズを再認識させます。VVR構成では、-fオプションが必要である事に注意してください。その後でvxprintコマンドでVxVMが認識しているブロックストレージのサイズを確認すると10-193-24-4\_disk\_1のサイズは20904976ブロック、すなわち10Gbyteで認識されています。

```
# vxdisk -g datadg -f resize 10-193-24-4_disk_1
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -
dm 10-193-24-4_disk_1 10-193-24-4_disk_10 - 20904976 - - -
:
(以下略)
:
```

同じように 2 号機でも、vxdisk コマンドに、ディスクグループ名である datadg と、再認識対象ディスク名の 10-193-34-4\_disk\_1 を指定して、ディスクサイズを再認識させます。VVR 構成では、-f オプションが必要である事に注意してください。その後で vxprint コマンドで VxVM が認識しているブロックストレージのサイズを確認すると 10-193-34-4\_disk\_1 のサイズは 20904976 ブロック、すなわち 10Gbyte で認識されています。

```
# vxdisk -g datadg -f resize 10-193-34-4_disk_1
[root@vcslin02 /]# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -

dm 10-193-34-4_disk_1 10-193-34-4_disk_7 < 20904976 - - - -
:
(以下略)
:
```

次に、1 号機（稼働系マシン）vradmin コマンドを使用し、ディスクグループ名、RVG 名、ボリューム名を指定し、ボリュームとファイルシステムのサイズを同時に 8Gbyte に拡張します。VVR 構成では vxresize ではなく、vradmin コマンドを稼働系マシンで実行すると、両ノードのボリュームが同時に拡張されます。

```
# vradmin -g datadg resizevol rvg01 datavol 8g
```

1 号機で vxprint コマンドを実行すると、datavol のサイズが 16777216 ブロック、すなわち 8Gbyte に拡張されたことが確認できます。

```
# vxprint
Disk group: datadg

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg datadg    datadg    -        -        -        -        -        -

dm 10-193-24-4_disk_1 10-193-24-4_disk_10 - 20904976 - - - -

rv rvg01      -          ENABLED -        -        ACTIVE -        -
rl rlk vcslin02-rep_rvg01 rvg01 CONNECT -        -        ACTIVE -        -
v datavol     rvg01     ENABLED 16777216 -        ACTIVE -        -
pl datavol-01 datavol   ENABLED 16777216 -        ACTIVE -        -
sd 10-193-24-4_disk_1-01 datavol-01 ENABLED 6291488 0 -        -        -
sd 10-193-24-4_disk_1-04 datavol-01 ENABLED 10485728 6291488 - -        -
pl datavol-02 datavol   ENABLED LOGONLY -        ACTIVE -        -
sd 10-193-24-4_disk_1-02 datavol-02 ENABLED 128 LOG -        -        -
v srl         rvg01     ENABLED 614400  SRL     ACTIVE -        -
pl srl-01     srl       ENABLED 614400  -       ACTIVE -        -
sd 10-193-24-4_disk_1-03 srl-01    ENABLED 614400  0       -        -        -
```

2号機でも vxprint コマンドを実行すると、datavol のサイズが 16777216 ブロック、すなわち 8Gbyte に拡張されたことが確認できます。

```
# vxprint
Disk group: datadg
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg	datadg	datadg	-	-	-	-	-	-
dm	10-193-34-4_disk_1	10-193-34-4_disk_7	-	20904976	-	-	-	-
rv	rvg01	-	ENABLED	-	-	ACTIVE	-	-
rl	rlk vcslin01-rep_rvg01	rvg01	CONNECT	-	-	ACTIVE	-	-
v	datavol	rvg01	ENABLED	16777216	-	ACTIVE	-	-
pl	datavol-01	datavol	ENABLED	16777216	-	ACTIVE	-	-
sd	10-193-34-4_disk_1-01	datavol-01	ENABLED	6291488	0	-	-	-
sd	10-193-34-4_disk_1-04	datavol-01	ENABLED	10485728	6291488	-	-	-
pl	datavol-02	datavol	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	10-193-34-4_disk_1-02	datavol-02	ENABLED	128	LOG	-	-	-
v	srl	rvg01	ENABLED	614400	SRL	ACTIVE	-	-
pl	srl-01	srl	ENABLED	614400	-	ACTIVE	-	-
sd	10-193-34-4_disk_1-03	srl-01	ENABLED	614400	0	-	-	-

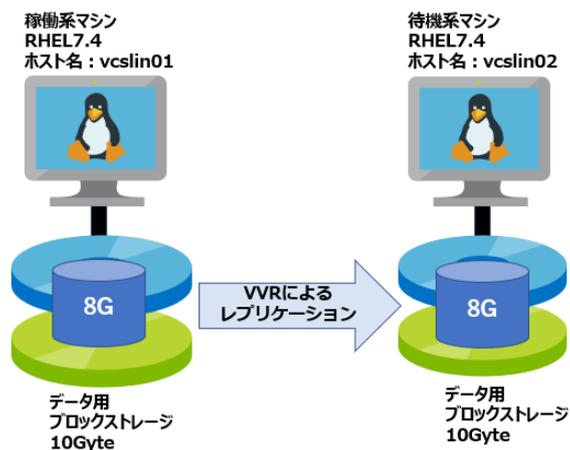
1号機で df コマンドで拡張対象のファイルシステム：/data のサイズが 8Gbyte に拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/fssdg/vol01 8.0G 38M 7.5G 1% /data
```

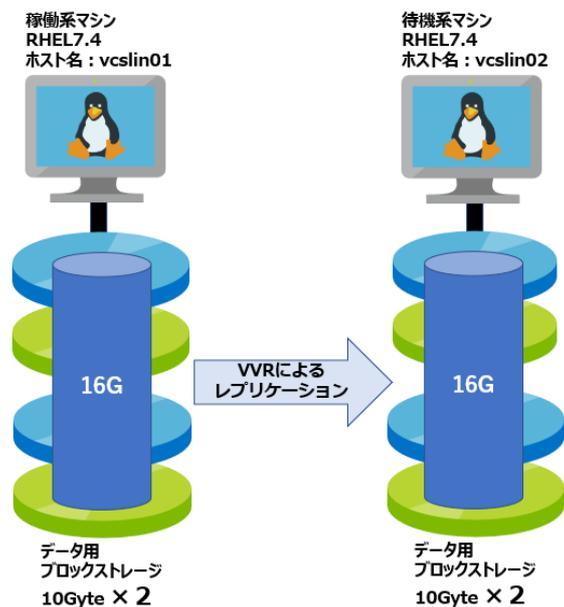
これで、作業終了です。

## 新規にブロックストレージを追加して、ボリュームを拡張するパターン

拡張前の構成は以下です。両ノードに、データ用として 10Gbyte のブロックストレージがそれぞれ接続されており、それをレプリケーションした 8Gbyte のデータ用ボリューム：vol01 が作成されています。データ用ボリューム：vol01 上にはベリタスファイルシステムが構成されており、稼働系マシンの /data にマウントされます。この状況下で、/data の容量を 16Gbyte に拡張する要件が発生したと想定します。



拡張後の構成は以下の通りです。10Gbyte ブロックストレージが追加され、それを使用してデータ用ボリュームとその上位のベリタスファイルシステムが同時に 16Gbyte まで拡張されます。



ここからは、実際の拡張作業を説明します。拡張前に、InfoScale のストレージ管理コンポーネントである VxVM および OS が認識しているディスクの状況を確認します。

1 号機で VxVM の管理下にあるディスクを確認します。VxVM の管理下であることを示す「STATUS が online」のディスクは 10-193-24-4\_disk\_1 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda              auto:none    -             -             online invalid
10-193-24-4_disk_4 auto:cdsdisk 10-193-24-4_disk_1 datadg        online
10-193-24-4_disk_7 auto:none    -             -             online invalid
```

次に、10-193-24-4\_disk\_1 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -             -             ENABLED
sdc              10-193-24-4_disk_4 10-193-24-4_disk_1 datadg        ENABLED
sdb              10-193-24-4_disk_7 -             -             ENABLED
```

さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。10Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc              8:32  0      10G  0 disk
├─sdc3           8:35  0      10G  0 part
└─sdc8           8:40  0      10G  0 part
```

念の為、拡張対象のファイルシステム：/data のサイズを確認します。8Gbyte である事が確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/datavol 8.0G  20M  3.8G  1% /data
```

同じ確認作業を 2 号機でも行います。2 号機で VxVM の管理下にあるディスクを確認します。VxVM の管理下であることを示す「STATUS が online」のディスクは 10-193-34-4\_disk\_1 の 1 つだけです。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda              auto:none    -             -             online invalid
10-193-34-4_disk_6 auto:none    -             -             online invalid
10-193-34-4_disk_7 auto:cdsdisk 10-193-34-4_disk_1 datadg        online
```

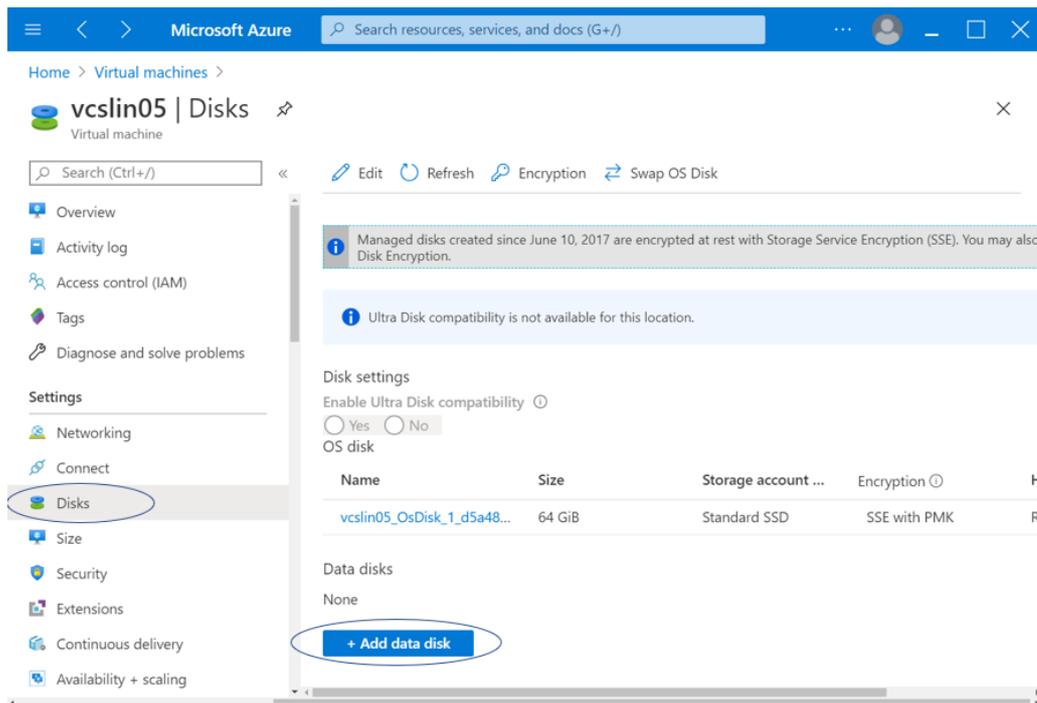
次に、10-193-34-4\_disk\_1 が OS からどのような名前でも認識されているか確認します。sdc で認識されている事が分かりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -             -             ENABLED
sdb              10-193-34-4_disk_6 -             -             ENABLED
sdc              10-193-34-4_disk_7 10-193-34-4_disk_1 datadg        ENABLED
```

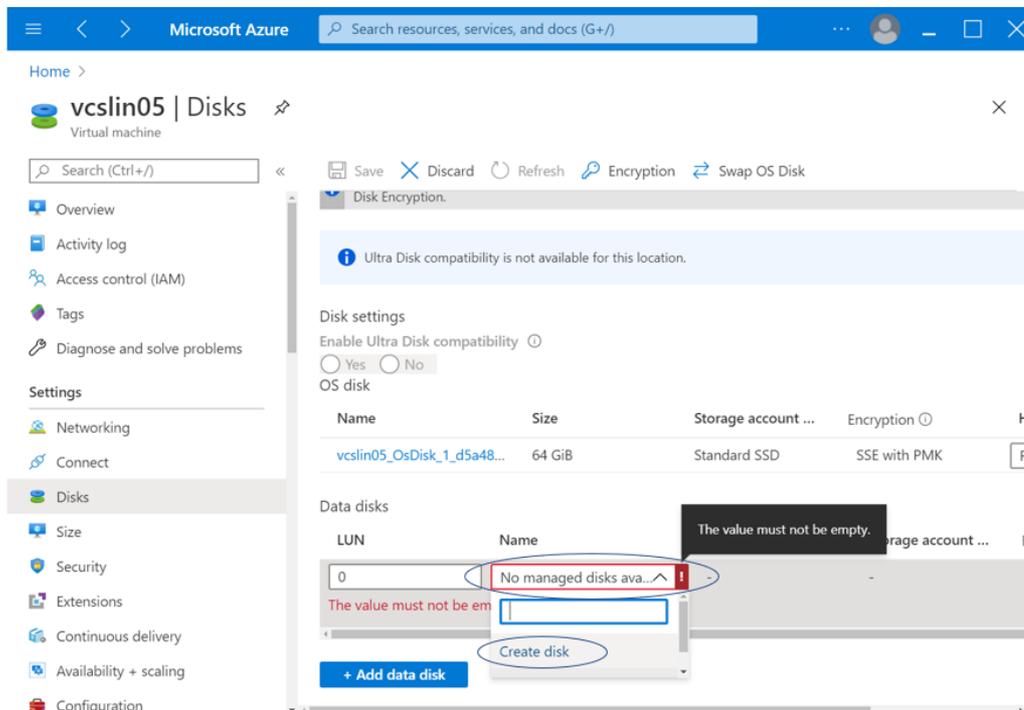
さらに、OS の lsblk コマンドを用い、sdc のサイズを確認します。10Gbyte である事が確認できました。

```
# lsblk | grep sdc
sdc          8:32  0    10G  0 disk
├─sdc3       8:35  0    10G  0 part
└─sdc8       8:40  0    10G  0 part
```

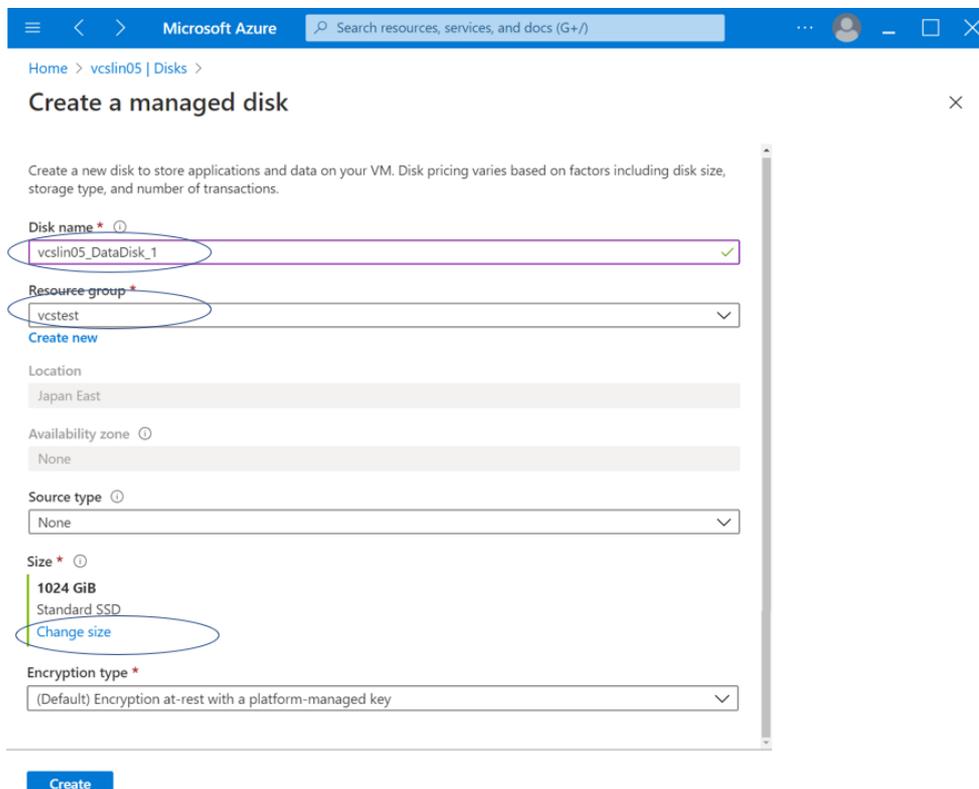
次に、Azure のコンソールで、ブロックストレージを追加します。ブロックストレージの追加は、仮想マシンを停止しないとできないので、インスタンスを停止してから下記の作業を行ってください。対象の仮想マシンのメニューで「Disk」を指定し、「Add data disk」をクリックしてください。



「Data disks」の欄の「Name」の下のプルダウンメニューを開くと「Create disk」が表示されますので、これをクリックしてください。



「Disk name」と「Resource group」を入力したら「Change size」をクリックしてください。



サイズを入力して「OK」をクリックしてください。

The screenshot shows the 'Select a disk size' dialog in the Microsoft Azure portal. The dialog has a search bar and a close button. On the left, there are navigation links for 'Home > v', 'Create', 'Resource group', 'Location', 'Availability', 'Source type', 'Size', and 'Encryption'. The main area contains a table of disk sizes and prices, and a section for creating a custom size.

Size	Price	Price	Price
128 GiB	\$10	500	60
256 GiB	\$15	500	60
512 GiB	\$20	500	60
1024 GiB	\$30	500	60
2048 GiB	\$40	500	60
4096 GiB	\$50	500	60
8192 GiB	\$60	1300	300
16384 GiB	\$70	2000	500
32767 GiB	\$80	2000	500

Below the table, there is a section for creating a custom size. It includes a text input field for 'Custom disk size (GiB) \*' with the value '10' entered. A green checkmark is visible next to the input field. Below the input field are 'Create' and 'OK' buttons.

内容を確認して「Create」をクリックしてください。

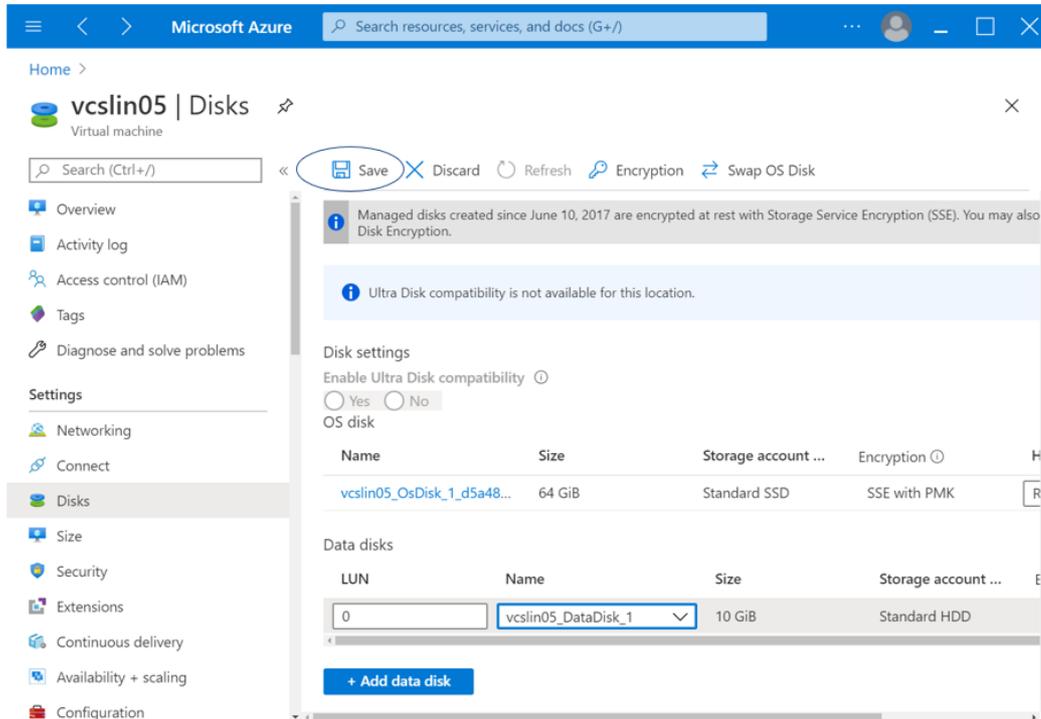
The screenshot shows the 'Create a managed disk' dialog in the Microsoft Azure portal. The dialog has a search bar and a close button. On the left, there are navigation links for 'Home > vmlin05 | Disks >'. The main area contains configuration options for a managed disk.

The configuration options are:

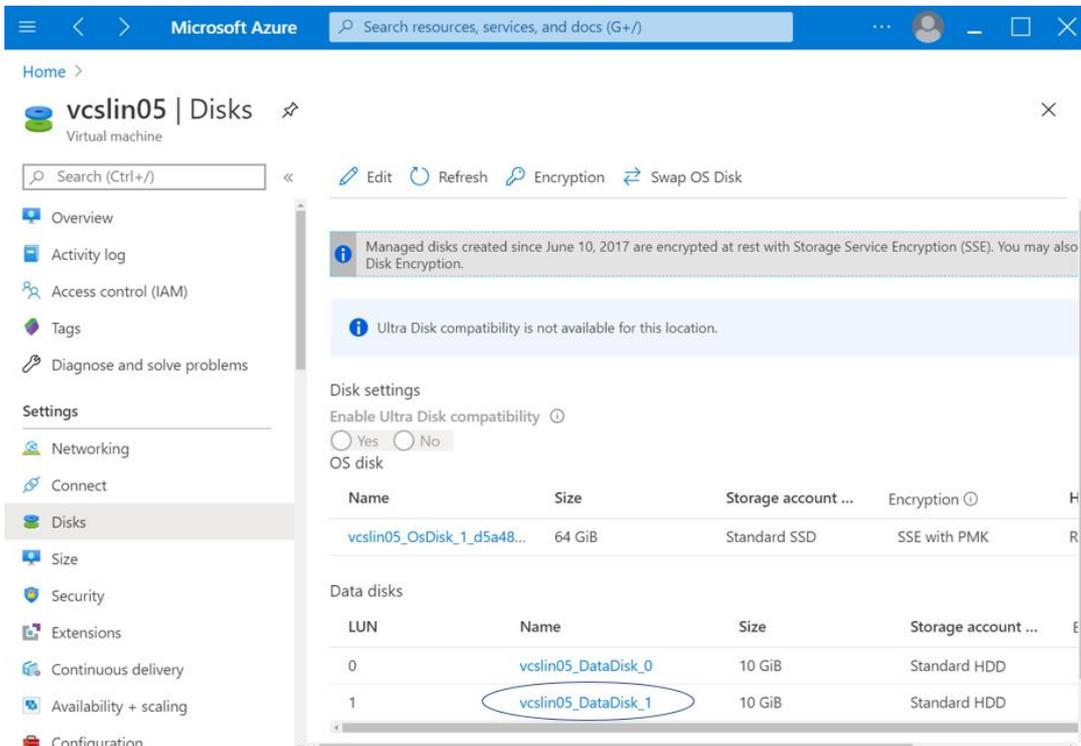
- Resource group \***: vcstest
- Location**: Japan East
- Availability zone**: None
- Source type**: None
- Size \***: 10 GiB (Standard HDD)
- Encryption type \***: (Default) Encryption at-rest with a platform-managed key

At the bottom of the dialog is a 'Create' button, which is circled in blue.

作成したブロックストレージが「Data disks」欄に表示されますので、Name やサイズなどの内容を確認の上「Save」をクリックしてください。



再度、対象の仮想マシンのメニューで「Disk」を指定すると、作成したブロックストレージが「Data disks」欄に表示されます。確認後、仮想マシンを立ち上げてください。尚、ここまでのブロックストレージ追加の作業はクラスター内の全ノードに対して行ってください。



1号機が立ち上がったら、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda                8:0    0   64G  0 disk
├── sda1            8:1    0   500M  0 part /boot/efi
├── sda2            8:2    0   500M  0 part /boot
├── sda3            8:3    0     2M  0 part
└── sda4            8:4    0    63G  0 part
sdb                8:16   0    50G  0 disk
└── sdb1            8:17   0    50G  0 part /mnt/resource
sdc                8:32   0    10G  0 disk
├── sdc3            8:35   0    10G  0 part
└── sdc8            8:40   0    10G  0 part
sdd                8:48   0    10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前でも認識されているか確認します。10-193-24-4\_disk\_9 という名前でも認識されている事がわかりました。

```
# vxdisk path
SUBPATH          DANAME          DMNAME          GROUP          STATE
sda              sda             -              -              ENABLED
sdb              10-193-24-4_disk_8 -              -              ENABLED
sdd              10-193-24-4_disk_9 -              -              ENABLED
sdc              10-193-24-4_disk_10 10-193-24-4_disk_1 datadg         ENABLED
```

vxdisk list コマンドで確認すると、10-193-24-4\_disk\_9 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-4_disk_8 auto:none    -             -             online invalid
10-193-24-4_disk_9 auto:none    -             -             online invalid
10-193-24-4_disk_10 auto:cdsdisk 10-193-24-4_disk_1 datadg         online
```

次に、vxdisksetup コマンドで 10-193-24-4\_disk\_9 をイニシャライズします。さらに vxdg コマンドで、既存のディスクグループ：datadg に、10-193-24-4\_disk\_9 を追加します。

```
# vxdisksetup -i 10-193-24-4_disk_9
# vxdg -g datadg adddisk 10-193-24-4_disk_9
```

再度 vxdisk list コマンドで確認すると、10-193-24-4\_disk\_9 の STATUS は online に変わって VxVM の管理下になり且つ既存のディスクグループ：datadg に所属した事がわかります。

```
# vxdisk list
DEVICE          TYPE          DISK          GROUP          STATUS
sda             auto:none    -             -             online invalid
10-193-24-4_disk_8 auto:none    -             -             online invalid
10-193-24-4_disk_9 auto:cdsdisk 10-193-24-4_disk_9 datadg         online
10-193-24-4_disk_10 auto:cdsdisk 10-193-24-4_disk_1 datadg         online
```

同じ作業を 2 号機でも行います。まず、lsblk コマンドを用いてディスクの状態を確認します。10Gbyte の sdd が追加されています。

```
# lsblk | grep sd
sda                8:0    0   64G  0 disk
├── sda1            8:1    0   500M  0 part /boot/efi
├── sda2            8:2    0   500M  0 part /boot
├── sda3            8:3    0     2M  0 part
└── sda4            8:4    0    63G  0 part
sdb                8:16   0    50G  0 disk
└── sdb1            8:17   0    50G  0 part /mnt/resource
sdc                8:32   0    10G  0 disk
├── sdc3            8:35   0    10G  0 part
└── sdc8            8:40   0    10G  0 part
sdd                8:48   0    10G  0 disk
```

次に、vxdisk path コマンドで sdd が VxVM からどんな名前でも認識されているか確認します。10-193-34-4\_disk\_8 という名前でも認識されている事がわかりました。

```
# vxdisk path
SUBPATH           DANAME           DMNAME           GROUP           STATE
sda                sda              -                -                ENABLED
sdb                10-193-34-4_disk_6 -                -                ENABLED
sdc                10-193-34-4_disk_7 10-193-34-4_disk_1 datadg          ENABLED
sdd                10-193-34-4_disk_8 -                -                ENABLED
```

vxdisk list コマンドで確認すると、10-193-34-4\_disk\_8 の STATUS は online invalid ですので、VxVM の管理下に無い事がわかります。

```
# vxdisk list
DEVICE           TYPE           DISK           GROUP           STATUS
sda              auto:none      -              -              online invalid
10-193-34-4_disk_6 auto:none      -              -              online invalid
10-193-34-4_disk_7 auto:cdsdisk  10-193-34-4_disk_1 datadg          online
10-193-34-4_disk_8 auto:none      -              -              online invalid
```

次に、vxdisksetup コマンドで 10-193-34-4\_disk\_8 をイニシャライズします。さらに vxvg コマンドで、既存のディスクグループ：datadg に、10-193-34-4\_disk\_8 を追加します。

```
# vxdisksetup -i 10-193-34-4_disk_8
# vxvg -g datadg adddisk 10-193-34-4_disk_8
```

再度 vxdisk list コマンドで確認すると、10-193-34-4\_disk\_8 の STATUS は online に変わって VxVM の管理下になり且つ既存のディスクグループ：datadg に所属した事がわかります。これで、追加したディスクを使用してストレージ領域を拡張する準備が整いました。

```
# vxdisk list
DEVICE           TYPE           DISK           GROUP           STATUS
sda              auto:none      -              -              online invalid
10-193-34-4_disk_6 auto:none      -              -              online invalid
10-193-34-4_disk_7 auto:cdsdisk  10-193-34-4_disk_1 datadg          online
10-193-34-4_disk_8 auto:cdsdisk  10-193-34-4_disk_8 datadg          online
```

1号機（稼働系マシン）vradmin コマンドを使用し、ディスクグループ名、RVG 名、ボリューム名を指定し、ボリュームとファイルシステムのサイズを同時に 16Gbyte に拡張します。VVR 構成では vxresize ではなく、vradmin コマンドを稼働系マシンで実行すると、両ノードのボリュームが同時に拡張されます。

```
# vradmin -g datadg resizevol rvg01 datavol 16g
```

1号機で vxprint コマンドを実行すると、datavol のサイズが 33554432 ブロック、すなわち 16Gbyte に拡張されたことが確認できます。

```
# vxprint
Disk group: datadg
```

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg datadg	datadg	-	-	-	-	-	-
dm 10-193-24-4_disk_1	10-193-24-4_disk_10	-	20904976	-	-	-	-
dm 10-193-24-4_disk_9	10-193-24-4_disk_9	-	20904976	-	-	-	-
rv rvg01	-	ENABLED	-	-	ACTIVE	-	-
rl rlk vcslin02-rep_rvg01	rvg01	CONNECT	-	-	ACTIVE	-	-
v datavol	rvg01	ENABLED	33554432	-	ACTIVE	-	-
pl datavol-01	datavol	ENABLED	33554432	-	ACTIVE	-	-
sd 10-193-24-4_disk_1-01	datavol-01	ENABLED	6291488	0	-	-	-
sd 10-193-24-4_disk_1-04	datavol-01	ENABLED	13998960	6291488	-	-	-
sd 10-193-24-4_disk_9-01	datavol-01	ENABLED	13263984	20290448	-	-	-
pl datavol-02	datavol	ENABLED	LOGONLY	-	ACTIVE	-	-
sd 10-193-24-4_disk_1-02	datavol-02	ENABLED	128	LOG	-	-	-
v srl	rvg01	ENABLED	614400	SRL	ACTIVE	-	-
pl srl-01	srl	ENABLED	614400	-	ACTIVE	-	-
sd 10-193-24-4_disk_1-03	srl-01	ENABLED	614400	0	-	-	-

2号機でも vxprint コマンドを実行すると、datavol のサイズが 33554432 ブロック、すなわち 16Gbyte に拡張されたことが確認できます。

```
# vxprint
Disk group: datadg
```

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg datadg	datadg	-	-	-	-	-	-
dm 10-193-34-4_disk_1	10-193-34-4_disk_7	-	20904976	-	-	-	-
dm 10-193-34-4_disk_8	10-193-34-4_disk_8	-	20904976	-	-	-	-
rv rvg01	-	ENABLED	-	-	ACTIVE	-	-
rl rlk vcslin01-rep_rvg01	rvg01	CONNECT	-	-	ACTIVE	-	-
v datavol	rvg01	ENABLED	33554432	-	ACTIVE	-	-
pl datavol-01	datavol	ENABLED	33554432	-	ACTIVE	-	-
sd 10-193-34-4_disk_1-01	datavol-01	ENABLED	6291488	0	-	-	-
sd 10-193-34-4_disk_1-04	datavol-01	ENABLED	13998960	6291488	-	-	-
sd 10-193-34-4_disk_8-01	datavol-01	ENABLED	13263984	20290448	-	-	-
pl datavol-02	datavol	ENABLED	LOGONLY	-	ACTIVE	-	-
sd 10-193-34-4_disk_1-02	datavol-02	ENABLED	128	LOG	-	-	-
v srl	rvg01	ENABLED	614400	SRL	ACTIVE	-	-
pl srl-01	srl	ENABLED	614400	-	ACTIVE	-	-
sd 10-193-34-4_disk_1-03	srl-01	ENABLED	614400	0	-	-	-

---

1号機でdfコマンドで拡張対象のファイルシステム：/dataのサイズが16Gbyteに拡張されたことが確認できます。

```
# df -h | grep /data
/dev/vx/dsk/datadg/datavol 16G 23M 15G 1% /data
```

これで作業は終了です。

---

## ベリタステクノロジーズについて

Veritas Technologies はエンタープライズデータ管理のグローバルリーダーです。複雑化した IT 環境においてデータ管理の簡素化を実現するために、世界の先進企業 50,000 社以上、Fortune 500 企業の 90 パーセントが、ベリタスのソリューションを導入しています。ベリタスのエンタープライズ・データサービス・プラットフォームは、お客様のデータ活用を推進するため、データ保護の自動化とデータリカバリを実現して、ビジネスに不可欠なアプリケーションの可用性を確保し、複雑化するデータ規制対応に必要なインサイトを提供します。ベリタスのソリューションは信頼性とスケーラビリティに優れ、500 以上のデータソースと 50 のクラウドを含む 150 以上のストレージ環境に対応しています。



## ベリタステクノロジーズ合同会社

<https://www.veritas.com/ja/jp>

〒107-0052 東京都港区赤坂 1-11-44 赤坂インターシティ 4F

ベリタスセールスインフォメーションセンター（法人のお客様向け製品購入に関する相談窓口）

■電話受付時間：10:00～12:00, 13:00～17:00（土、日、祝日、年末年始を除く）

■電話番号：0120-907-000（IP 電話からは 03-4531-1799）

© 2019 Veritas Technologies LLC. All rights reserved. Veritas および Veritas のロゴは、米国およびその他の国における Veritas Technologies LLC またはその関連会社の商標または登録商標です。その他の名称は、それぞれの所有者の商標である場合があります。