Lessons Learned Building Db2 Clusters on Amazon AWS

Philip Nelson

ScotDB Limited

(teamdba@scotdb.com)

Agenda

- Introduction to cloud technology
- Different ways to run Db2 on the cloud
- Why “roll your own” on top of IaaS
- Lessons learned about Db2 features while setting up on AWS
- Lessons learned about AWS (and other cloud providers)

Cloud Technology Overview

Is it “just somebody else’s computer?”

Cloud Services

COMPUTE RESOURCES

STORAGE

NETWORKING

SPECIALIZED SERVICES

Compute Resources

Virtual machines (VM)

- Different operating systems and resource sizing available

Managed Kubernetes clusters

Managed Docker environments

Serverless computing

- Doesn’t mean there are no servers involved : just not configured by you

Storage Facilities

Object storage

- Storing files that are not accessed internally e.g. backups, logs, images

File storage

- File system that can be mounted in multiple places (similar to NFS)

Block storage

- Like “disks” that can be formatted with file systems and mounted on servers

Memory cache

- For temporary files that need quick access

Networking

- Internal private networks between cloud services
- Links between cloud services and external services and the Internet
- Private connections between cloud and on-premises services
- Domain management services

Specialized Services

- Functionality to perform a particular task or business function
- Normally accessed through an API
  - May also provide a browser based front end
- Enable the use of complex services without specialist knowledge
- Many of these services are data related
- Examples –
  - Machine Learning
  - Specialized analytics
  - Multimedia processing

Different Ways to Run Db2 in the Cloud

IBM DBaaS offerings

- Db2 Warehouse on Cloud
- Db2 on Cloud

Containerized

- Red Hat Openshift
- Other Kubernetes
- Docker

IaaS (aka “roll your own”)

Why “Roll Your Own” on top of IaaS?

Maximize control over environment

- Happy with cloud provider’s operating system installation
- Would like to specify your own Db2 installation and layout
- Need maximum flexibility in backup and recovery options
- Want to configure a Db2 environment not yet available as a packaged offering

Useful for learning / experimentation

- Provides environments without high cost / long lead times

Lessons Learnt About Db2 While Setting Up on Amazon AWS

- Which ports (TCP, UDP and ICMP) are needed for Db2 clusters?
- How can Db2 exploit cloud object storage?
- Which versions of Linux will let you install TSA/MP?

Db2 Port (Network Security) Requirements

Port Type	Port Number	Description
TCP	50000	Db2 standard port (default : configurable)
TCP	50443	Db2 TLS (SSL) port (configurable)
TCP	55555	Db2 database HADR transmission port (configurable : one per DB)
TCP	657	TSA port 1 (TCP)
UDP	657	TSA port 1 (UDP)
UDP	12347	TSA port 2 (UDP)
UDP	12348	TSA port 3 (UDP)
ICMP	-	TSA ping (ICMP)

Db2 and Cloud Object Storage

- More details on Cloud Object Storage (specifically Amazon AWS S3)
- Linking Db2 to Amazon AWS S3
- Uses for AWS S3 in Db2
- Futures

Amazon AWS S3 (Cloud Object Storage)

S3 = Simple Storage Service

Stores files as “objects” : can only be read / written in entirety

Arranged into “buckets”

- Unique across AWS
- Public or private
- Encrypted or unencrypted

Free to upload, pay for storage and downloads (outside AWS region)

Linking Db2 to AWS S3

Provide	Provide a PKCS encrypted keystore
Define	Define Storage Access Alias
Use	Use Storage Access Alias with Db2 Utilities

Providing a PKCS Encrypted Keystore

To access S3 requires two credentials –

- Access Key ID (in Db2 documentation as “storage-user-ID”)
- Secret Access Key ID (in Db2 documentation as “storage-password”)

Stored in the same keystore used by Db2 native encryption -

- Typically stored in ~db2inst1/security
- `gskcapicmd_64 -keydb -create -db native.p12 -pw “<strong-pw>” -type pkcs12 -stash -strong`
- `db2 “update dbm cfg using keystore_location /home/db2inst1/security/native.p12 keystore_type pkcs12”`
- `db2stop`
- `db2start`

Define Storage Access Alias

```
>>- CATALOG STORAGE ACCESS ALIAS <alias> →
>- VENDOR S3 SERVER --+- DEFAULT ----+→
                      +- <endpoint> -+
>- USER <storage-user-ID> →
>- PASSWORD <storage-password> →
>-+-----------------------------------+→
  +- CONTAINER <container-or-bucket> -+
>-+-------------------+-+----------------------+ →←
  +- OBJECT <object> -+ +- DBGROUP <group-ID> -+
                        +- DBUSER <user-ID> ---+
```

Notes on Storage Access Aliases

Cataloged at instance level

- Potentially available to any database

By default only accessible by SYSADM

- Parameters DBGROUP and DBUSER allow a group or user to be given access

Default region is “US East 1 Virginia”

- Important for GDPR considerations to using non-default endpoint
- http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region

CONTAINER parameters allows access to specific buckets in region

“LIST STORAGE ACCESS” command lets you list registered aliases

Examples of Alias Creation

Bucket Level

```
CATALOG STORAGE ACCESS ALIAS
scotdbeu VENDOR S3 SERVER
s3-eu-west-1.amazonaws.com
USER <access-key-ID>
PASSWORD <secret-access-key-ID>
CONTAINER scotdb.scotdbeu
```

Region Level

```
CATALOG STORAGE ACCESS ALIAS
s3euw1 VENDOR S3 SERVER
s3-eu-west-1.amazonaws.com
USER <access-key-ID>
PASSWORD <secret-access-key-ID>
```

Uses for AWS S3 in Db2

Writing output from BACKUP and LOAD COPY

Loading data directly from S3

Exporting data directly to S3

All refer to location using DB2REMOTE syntax

- DB2REMOTE://bucket-alias>//<storage-path/<file-name>
- DB2REMOTE://<region-alias>**/<bucket>/**<storage-path>/<file-name>
- <file-name> only not used for BACKUP and LOAD COPY

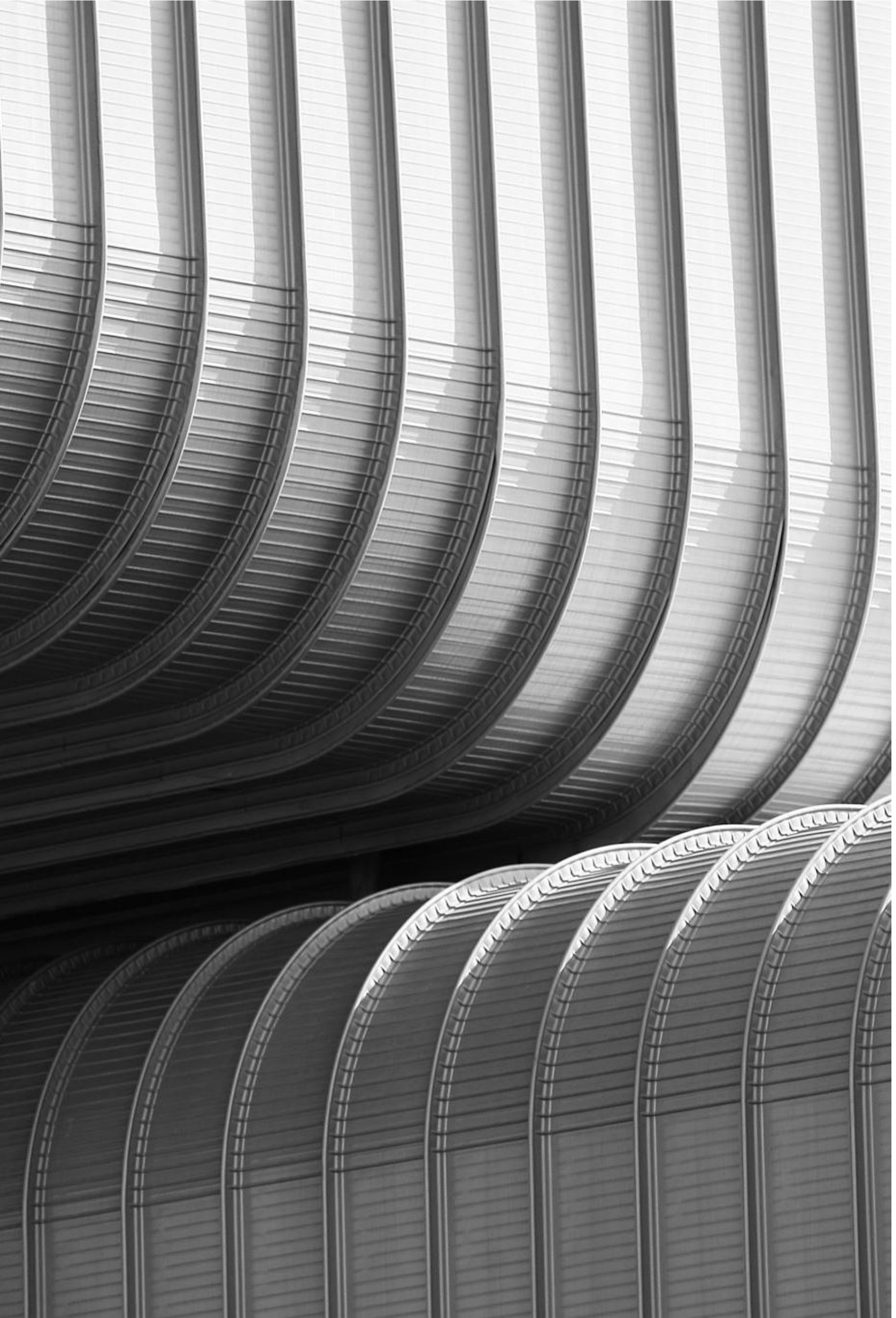
Examples

Backup database to bucket and region aliases (equivalent) –

- BACKUP DATABASE SAMPLE TO DB2REMOTE://scotdbeu//DB2Backups
- BACKUP DATABASE SAMPLE TO DB2REMOTE://s3euw1/scotdb.backups/Db2Backups

Loading from a bucket alias –

- LOAD FROM DB2REMOTE://scotdbeu//loadfiles/loaddata.ixf OF IXF REPLACE INTO <myschema>.<mytable> …



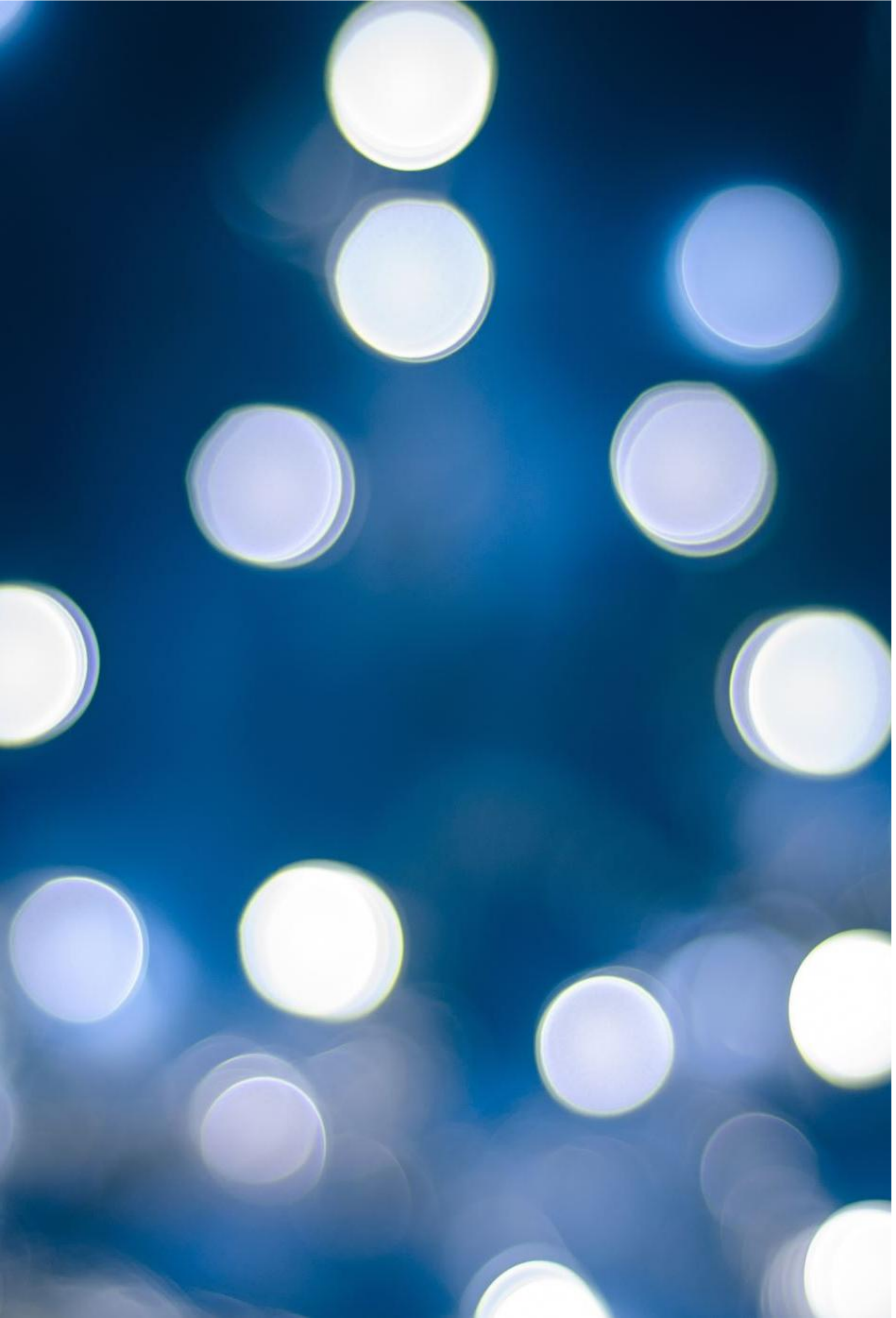
Further Notes

For utilities producing content (BACKUP, LOAD COPY and EXPORT) a local staging directory is used

- Default is <instance-home>/sqllib/tmp/RemoteStorage.<xxxx>
- Change via Db2 registry variable
  - DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH

S3 buckets appear to be limited to 5 GB in size

- Backup is split into multiple files of this size automatically
  - Uses same naming standard as for backing up to multiple local directories / TSM streams
- Newer bucket type (recently added to AWS) allows larger size
  - Db2 doesn’t seem to exploit this yet



Db2 Object Store Futures …

Currently not possible to archive logs to object storage

- This would complete the picture for high availability

Aha! Idea (created by me) to add this –

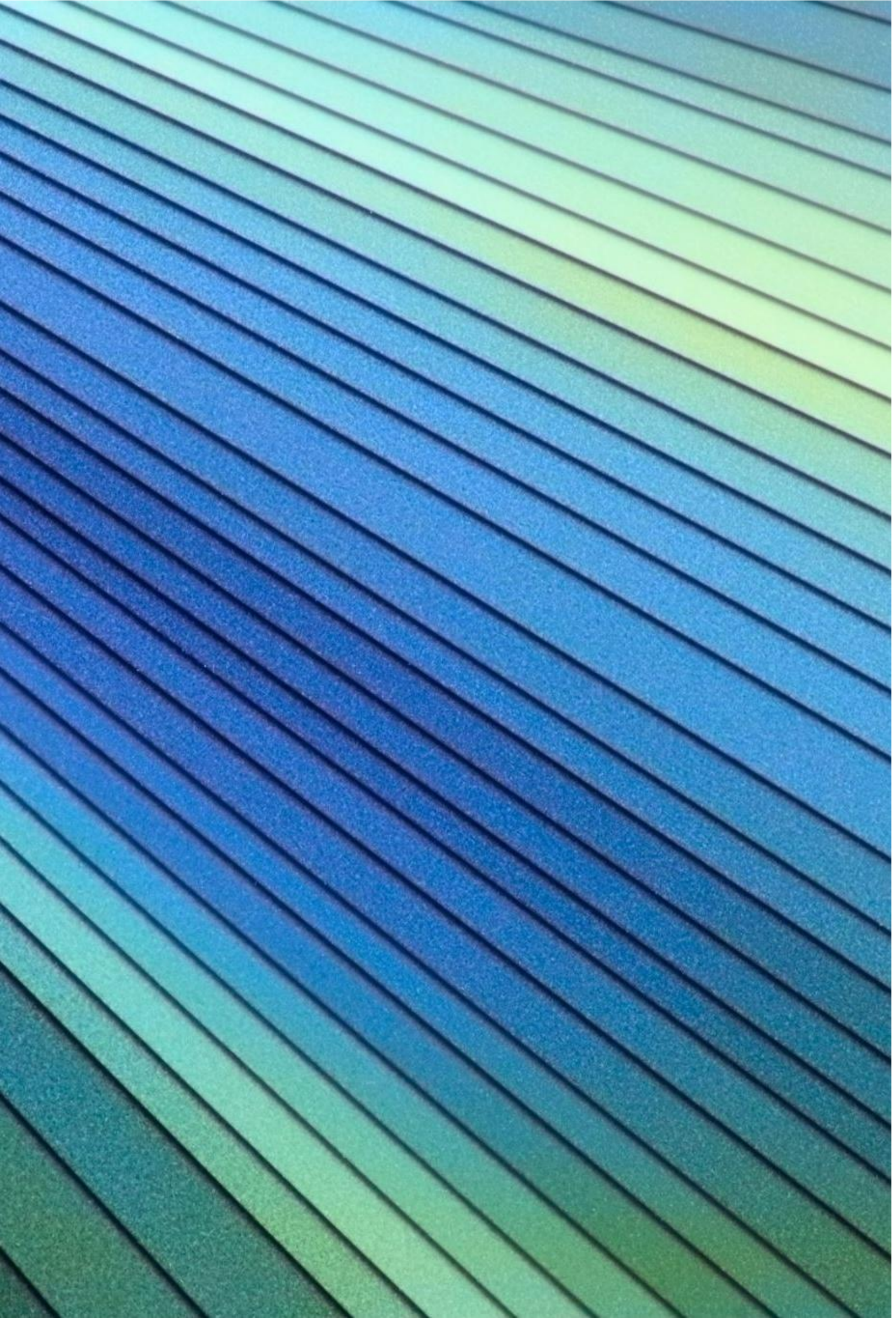
https://ibm-data-and-ai.ideas.aha.io/ideas/DB24LUW-I-862

Recently (29th April 2021) changed status to "Future Consideration"

- Publicly mentioned by IBM on social media as "under construction"

Hints of more object store usage in 11.5.7.0 (due imminently)

- Nothing released publicly in the Db2 EAP beta stream yet however!!!



Linux and TSA/MP

Db2 uses TSA/MP as cluster manager for automated failover

- Being superseded by PaceMaker

TSA/MP code not developed by Db2 team

Shipped as RPMs (RedHat Package Manager packages)

- DEBs (Debian / Ubuntu) provided – but not used by automated installation

TSA/MP installation script very “picky” about Linux distro / version

By installing RPMs or DEBs manually, more choice is available

- Have tested installation on CentOS (RPMs) and Ubuntu 18.04 LTS (DEBs)
- Not yet managed to test a full cluster configuration

Lessons Learnt About AWS While Setting Up Db2

- What is the scope for moving a Virtual IP?
- How can you incur large costs in a hybrid (cloud / on-premise) setup?
- How do you allocate separate disks for Db2 data, logs and backups?
- How do you share a disk between AWS Virtual Machines?
- How do you automate cluster builds?

Scope for a Virtual IP : Why?

- Fully automated Db2 HADR clusters use a Virtual IP
- Defined during TSA/MP (cluster) setup
- Moved during failovers (planned our unplanned)

Virtual IP : Alternatives

- ACR (Automated Client Routine) : for current active connections
- db2 “UPDATE ALTERNATE SERVER FOR DATABASE USING HOSTNAME <x> PORT <n>”
- Client configuration : JDBC
  - ClientRerouteAlternateServerName
  - ClientRerouteAlternatePortNumber
- Client configuration : non-JDBC (db2dsdriver.cfg parameters)
  - Add <alternateserverlist> stanza
  - Set parameters enableAcr and enableAlternateServerListFirstConnect
- DNS change for hostname → IP address mapping
  - Services in AWS to both monitor hosts and change DNS (Route53)

Costs To Beware of : Why?

- Wrong decisions about how to use Db2 with AWS can have large costs
- Some of the biggest costs are not in obvious places
- Risk of large costs increase when –
    - Operating in a hybrid (part cloud and part on-premise) environment
    - Operating across AWS regions

Costs to Beware Of : Data Transfers

Summary of pricing model -

- Data transfers into AWS are free
- Transfers between AWS regions incur a cost
- Transfers out of AWS incur a greater cost

Db2 setup that could prove very costly

- Frequently restoring backups from AWS S3 to on-premise or another region
- HADR log shipping from AWS to on-premise or another region
- Loading data from AWS S3 to on-premise or another region
- On-premise application servers with database servers in AWS

COLOCATION IS KING!!!

Costs to Beware Of : Elastic IP Addresses

Most resources in AWS are only charged when active

- e.g. Shut down a server and all you pay for is the storage

Elastic IP addresses (public facing static IP addresses) are different

- Static IP addresses are a limited resource
- To encourage good practice EIPs are charged as long as they are allocated

Therefore only define an EIP if you need a public facing address

- Use internal addresses for Db2-to-Db2 and appserver-to-Db2 connections
- AWS generated dynamic IPs for VMs by default – fine for admin purposes
- Investigate AWS networking capabilities for more options

How To Use Separate Disks : Why?

- Best practices for storage layouts still apply in the cloud
- Typically separate storage for (at least) –
  - Data
  - Logs
  - Backups
- Provides benefits in terms of –
  - Resilience
  - Performance e.g. parallelism
  - Cost e.g. use better storage where it matters
  - Ease of administration
    - Adding storage to certain areas more easily
    - Taking snapshots of some disks may benefit during cloning of systems

Typical Layout For a Small Db2 Server

Device	Size (GB)	Used For
/dev/sda	20	/ (operating system)
/dev/sdb	20	/opt/IBM (Db2 software)
/dev/sdc	20	/db2log11 (primary logging)
/dev/sdd	20	/db2log12 (secondary logging)
/dev/sde	25	/db2data11 (data – 1 of 4)
/dev/sdf	25	/db2data12 (data – 2 of 4)
/dev/sdg	25	/db2data13 (data – 3 of 4)
/dev/sdh	25	/db2data14 (data – 4 of 4)
/dev/sdi	20	/db2home (Db2 instance home)
/dev/sdj	200	/db2bck11 (Db2 backups)

How To Configure Disks in AWS (1)

- By default servers get created with one disk : /dev/sda
  - This can be resized as appropriate (if adding multiple disks often reduce)
- Use “Add Volume” to add each required disk
  - Device name automatically allocated, so just set size
- Launch instance
- Check disks are available for use
  - sudo lsblk –f : added /dev/sd<x> will show as /dev/xvd<x>
- Prepare file systems on each disk
  - sudo mkfs –t xfs /dev/xvd<x>

How To Configure Disks in AWS (2)

- Create mount points in the file system for each disk
    - sudo mkdir /<mount-point>
- Mount each disk into the file system
    - sudo mount /dev/xvd<x> /<mount-point>
- Prepare for auto mounting
    - sudo blkid : gets list of UUID values
    - sudo cp /etc/fstab /etc/fstab.orig : take backup of fstab
    - Edit /etc/fstab adding lines of format -
  - UUID=<uuid> /<mount-point> xfs defaults,nofail 0 2
    - sudo umount /<mount-point> : for each disk
    - sudo mount –a : mount all disks automatically
- DO NOT LEAVE THE SERVER UNLESS “mount –a” WORKS WITHOUT ERROR
    - If necessary restore /etc/fstab back from /etc/fstab.orig

Disk Configuration : Further Notes

- Volumes can be resized easily
  - Offline and now also possible to do some changes online
- Other changes are also easy
  - e.g. switching to a server with more (or less) resources

How to Share Disks Between VMs : Why?

- Both servers in an HADR pair need to access LOAD COPY files
    - If not available, and file not copied over quickly, HADR will fail
- Also useful when preparing HADR standby
    - Backup files can be accessed from both servers
- On-premise servers use one of two mechanisms
    - TSM (Spectrum Protect) nodes with shared aliases
    - NFS mounted file systems

How to Share Disks Between Servers

- AWS provides Elastic File Storage (EFS)
  - Uses NFS protocols
- Create EFS file system
  - In AWS Console, access Storage/EFS and choose “Create File System”
  - Associated with appropriate VPC and security groups
- Install necessary software on VM –
  - sudo yum install nfs-utils
- Mount EFS volume on directory (assumes /db2bck11 unused)
- mount –t nfs4 –o
- nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
- <*efs-url*>:/ db2bck11

How To Automate Cluster Builds – Why?

- Setting Up Even a Single Cluster Is Time Consuming
- Many “clicks” (in AWS console) and “keystrokes” (from command line)
  - Easy to miss something
- Db2 already has automated installation and configuration facilities
  - db2setup –r <response-file>
  - Choice of Db2 command line tools (clp, clpplus)
  - db2haicu –f <config-file>

How to Automate Cluster Builds

- AWS supplies a build automation tool called CloudFormation
  - YAML or JSON formats
  - Graphical tool available
  - Allows parameterization
- AWS blog on configuring Db2 cluster with CloudFormation –
- https://aws.amazon.com/blogs/database/creating-highly-available-ibm-db2-databases-in-aws/
- Some comments –
  - Sample build installs on top of Windows!!!
  - It is for Db2 V11.1
  - It doesn’t always follow best practices (Db2 or AWS)
  - But it is a good starting point … ANY VOLUNTEERS ???

Speaker: Philip Nelson
Company: ScotDB Ltd.
Email Address: teamdba@scotdb.com

Please provide feedback!