MIGRATING SQL SERVER QUICKLY AND EFFICIENTLY

INTRODUCTION

- Sr. DBA at Alberta Gaming & Liquor Commission
- Originally from Kamloops BC
- 10 years of IT experience
  - 6 as a SQL Server DBA
  - Contractor for last 3 years
- Worked with SQL Server 2000 through 2014
  - Experience with other RDBMS such as Oracle, MySQL
- MCSA: SQL Server 2012
- Email: Ian.A.Chamberland@gmail.com

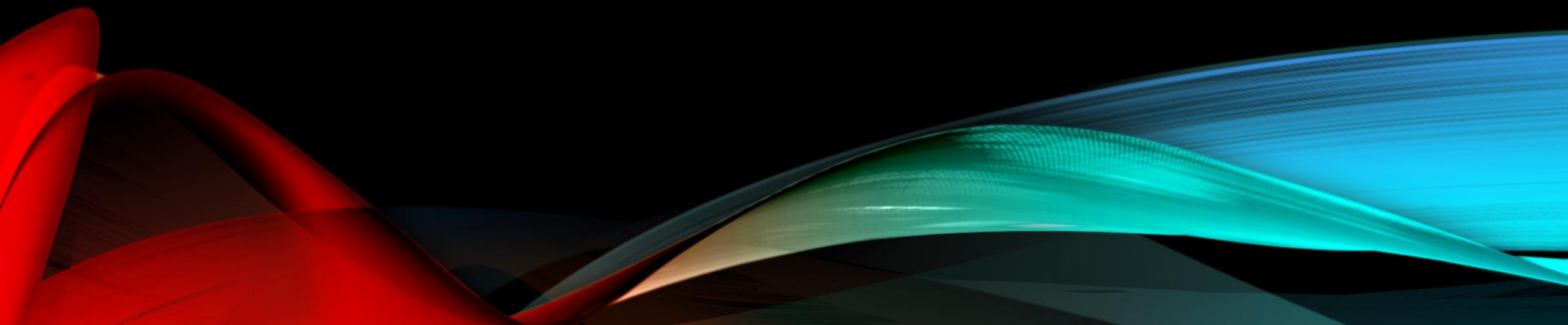
HOW DID THIS PRESENTATION COME ABOUT?

- Various projects requiring some form of database migration
- Projects included
  - Hardware refresh
  - SQL Upgrade
  - SAN refresh
- Failover cluster with 8 SQL instances
  - Approximately 200 databases total
- Lots of different applications including
  - Jobs/SSIS
  - SSRS Reports/Tableau/Excel
  - Services, Web Apps, desktop client
- Outage windows usually 15 minutes but 1 hour could be arranged

AGENDA

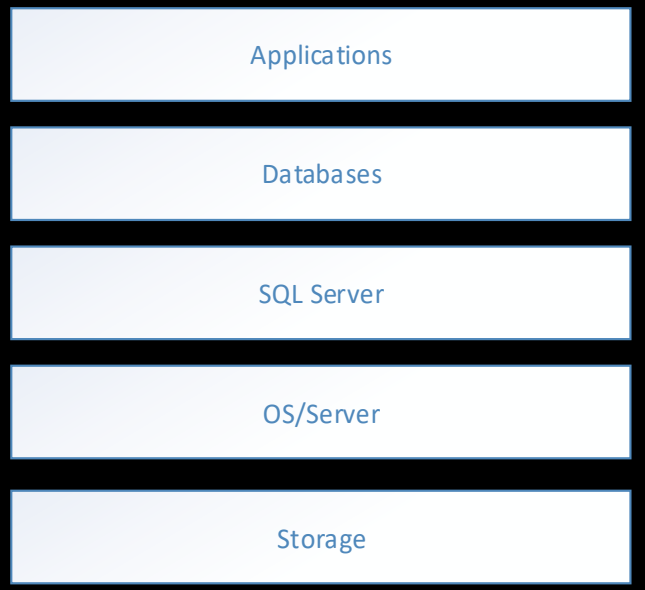
- Analysis
  - What is changing
  - What is the impact
- Planning
  - Instance
  - Database
  - Application
- Execution

ANALYZING THE IMPACT



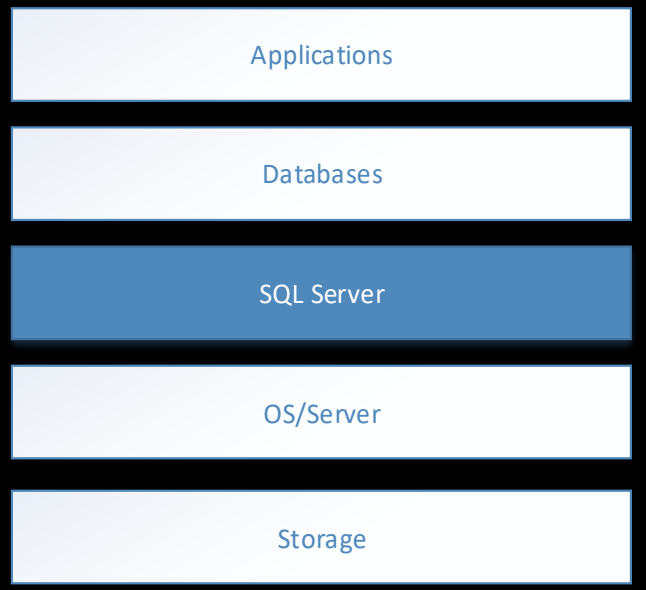
WHAT IS BEING MIGRATED?

- Application Upgrades
- SQL Server Upgrades
- Windows Upgrade
- Server Refresh
- SAN Refresh



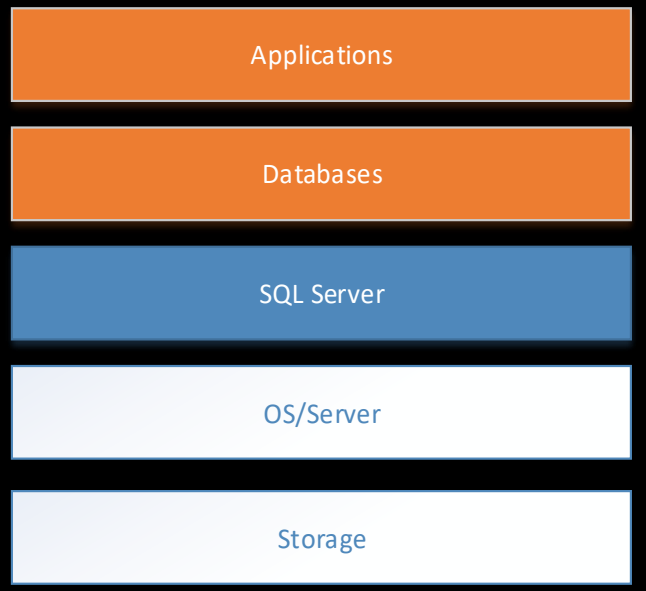
MIGRATION SCENARIO

- Migrating from 2008R2 to 2014
- It's a two node cluster
  - Over 200 databases
- Move everything in one window
- Outage window is one hour



WHAT IS THE IMPACT

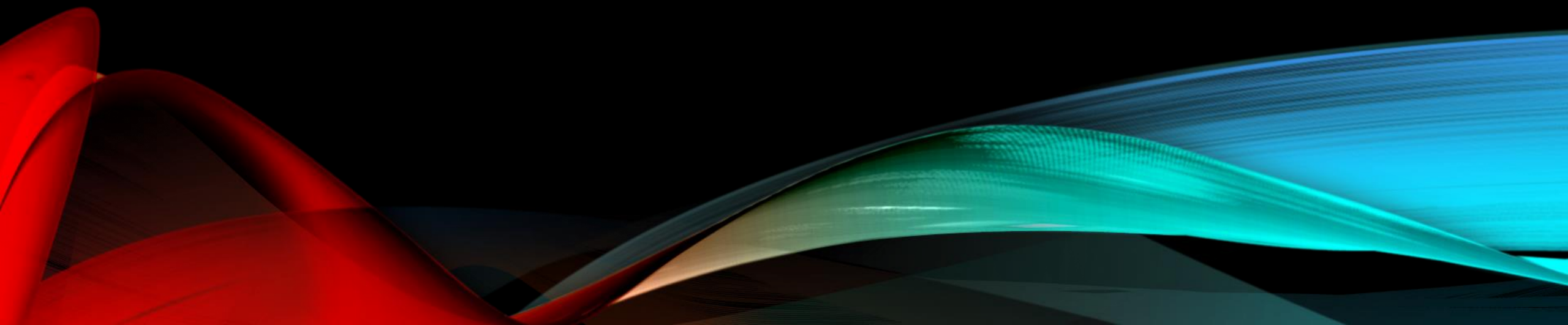
- What databases need to be migrated
- Instance Objects
- What applications are affected?
- Connection string changes



WHAT TOOLS SO WE HAVE?

- What tools can we use to identify the impact?
  - Profiler/Extended Events
  - Login Auditing
    - Set for “both failed and successful logins”
  - Auditing Tools
  - System Documentation
  - Application Testing

CREATING A PLAN



INSTANCE SETUP

- Install and Configure the new instance
- Create instance objects
  - Jobs
  - SSIS Packages
  - Maintenance plans
  - Linked Servers
- How to migrate
  - SSIS
  - Scripts (T-SQL, PowerShell)
- Demo

USER DATABASES

- How can we prepare before hand?
- Use HA/DR technologies in SQL Server
  - Mirroring
  - Log Shipping
  - Availability Groups
- Script your own process
  - Backup/Restore
  - Detach/Copy/Attach

MIRRORING

- Simple switch over
  - ALTER DATABASE <DB>
    SET PARTNER FAILOVER;
- Synchronous commit ensures databases are in sync
- Requires Full Recovery Mode
- Deprecated Technology
- Only one destination
- Built into SQL Server (2005+)

MIRRORING

1. Setup Mirroring
   - Do this earlier in the week.
   - Set asynchronous
2. Set mirroring to synchronous
   - This may create a performance impact
   - Perform this step closer to the migration
   - Monitor Closely
3. Mirroring Failover
   - Flip the mirror partner
   - Partner is now the primary
4. Tear Down
   - Cleanup any leftover mirroring setup

AVAILABILITY GROUPS

- Similar to mirroring
  - Simple switchover
  - Synchronous replication
- Requires Full Recovery mode
- Enterprise Feature (2012 & 2014)
- Require Windows Clustering
- With listener applications connections will follow
- SQL Server 2016
  - Distributed Availability Groups
  - Basic Availability Groups

LOG SHIPPING

- More moving pieces
- Switch over more complicated then mirroring
  - Backup and Restore tail log
- Requires Full Recovery mode
- Multiple Destinations
- Built into SQL Server

LOG SHIPPING

1. Setup Log Shipping
   - Monitor to ensure log restores are keeping up
2. Disable Log backups
   - Hour before Migration
   - Monitor to ensure log restores are completed
3. Migration
   - Backup log with no recovery
   - Restore with recovery
4. Post Migration
   - Remove Log Shipping

BACKUP AND RESTORE

- Potentially slower process
- Databases can be in simple recovery
- Can restore full ahead of time
  - Then use a differential to catch up
- Greater control with custom scripts
- Dependent on backup strategy

BACKUP AND RESTORE

1. Restore Full Backup
   - If using Differential Backups can be done days before
   - Can use Copy Only Backups to preserve differential backup chain
2. Take Differential Backup
   - Can switch to full recovery if using simple
3. Restore Differential Backup
4. Restore Logs
   - Check MSDB Database for new backups
   - Create Restoration Job
5. Backup Tail Log
   - BACKUP LOG <DB> WITH NO RECOVERY
6. Restore Tail Log
   - RESTORE LOG <DB> WITH RECOVERY

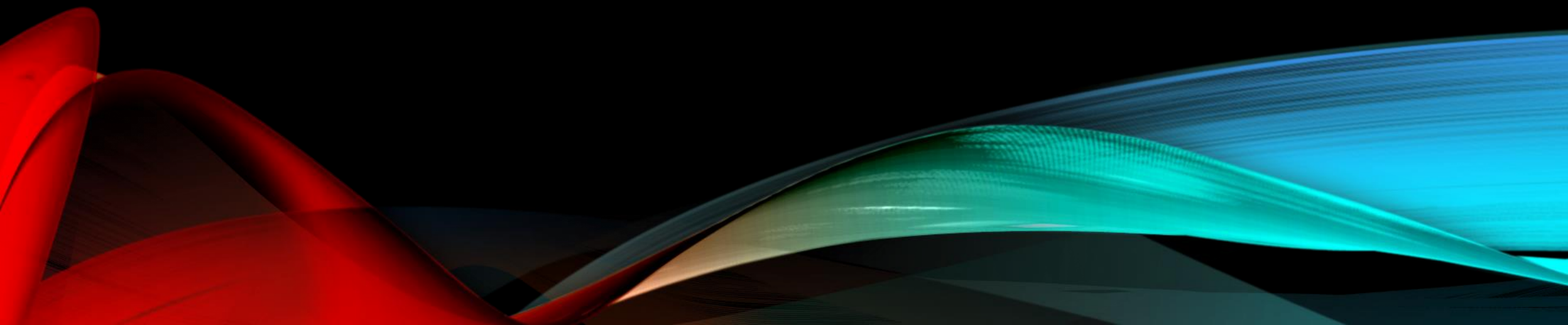
FAILOVER CLUSTERING

- Viable if…
  - If SAN stays the same
  - Already set up with a FCI
- Migration to a new server is just a cluster flip
- New Server can have upgraded version of SQL Server
  - Inline upgrade
  - Limited Roll back plan
- Windows Server 2016
  - Rolling Upgrade of Failover Clusters

APPLICATIONS

- How can we control the application changes
- Client Configuration
  - Connection string changes
  - SQL Server Alias
  - System DSN
- Central Configuration
  - DNS CNAME
  - Re-Use DNS/IP Changes
  - Centrally stored app configuration? (CIFS Share)
  - AlwaysOn Availability Group Listener

EXECUTE THE MIGRATION



PREPARATION

- Use T-SQL/PowerShell scripts where ever you can
- Document the steps
  - Create a checklist for each instance
- Practice the migration
  - In Dev\Test walk through the migration several times
  - Find out how long you need
  - Refine the process
- Test the applications
  - Was anything missed in the analysis?
  - Are there any surprises

EXECUTION

- Get your outage window right
- Ensure support staff are on hand

CONCLUSION

QUESTIONS?

Email: Ian.A.Chamberland@gmail.com

Demo Code: https://github.com/Ichamberland/