



Managing The Appliance

Agenda

- Configuring the Appliance
- Trusted Authentication
- Instant File Initialisation
- Transparent Data Encryption
- Permissions Model
- Workload Management
- Backup & Restore
- Monitoring

Configuring the Appliance

PDW Region Domain Administrators

Domains

- Fabric Active Directory
- PDW Active Directory

Creating New Domain Admins

Create new User

- Log into domain virtual machine
- Use dsa.msc to add a user

Add new user to following groups

- Domain Admins
- PDWControlNodeAccess

Password

- Use dwconfig.exe on Management Node to reset
- Don't include \$ in the password

HDInsight Domain Administrators

- Single account called Administrator Password Reset
- dwconfig.exe

It is not recommended to add other users into Domain Admins group in the Hadoop Region

Chicken & Egg

What if?

- Domain Admin password has expired
- No active session on management node

Answer

- Reset password on the login screen
- Reset password again via dwconfig
 - Ensures password is properly propagated and account is supported

Additional Accounts

- Polybase – pdw_user in Hadoop
- System Center – Region logins required
- Backup / Restore – Customer domain user

Password also needs
to be set for
Directory Services
Restore Mode for
active directories

DNS Forwarding

Required for

- Windows Update
- Polybase (Azure)

Set via Powershell

- RDP onto MAD01
- Set-DnsServerRecursion
–Enable \$True
- Add-DnsServerForwarder
–IPAddress <IP>
–PassThru

Externally Trusted Certificates

Required for

- Management Console
- Developer Dashboard
- Hive ODBC Connector

Trusted Authentication

In Scope

Protocols

- Kerberos
- NTLM
- SSPI (API)

* Partial support for Management Console as credentials need to be submitted a second time

Surface Area

- dwloader
- SSIS Destination Adaptors
- SSDT
- SQLCMD
- Management Console*

Out of Scope

- Hadoop Region
- Non Windows Domains
- Users outside of corporate identity management systems (Internet)
- Un-Trusted Domains
- Propagation of security Context
 - To Compute Nodes
 - For Polybase
 - Backup / Restore
- Trusted Auth Only Authentication Configuration

Out of Scope – Client Tools

- Informatica Connector
- BO Connector
- DWConfig
- PAV

Important Notes

- Trust is between Corporate Domain and Workload Domain
- Minimum Configuration (NTLM)
 - 1 way (outgoing) external (non-transitive) trust between corporate DC and PDW Workload AD
- Minimum Configuration (Kerberos)
 - 1 way forest
 - 2 way forest
 - 2 way external Trust

Instant File Initialisation

Security Measure

- Windows Policy
 - Perform Volume Maintenance Tasks
- Enabled prevents PDW having to zero-out pre-sized data files
- Disabled requires PDW to zero out data files
- Transaction log file is always zeroed out

Configuration

- Set by dwconfig.exe
- Stop & Start appliance required to take effect

Transparent Data Encryption

TDE = Encryption at rest

Encryption of data at the page level

- Encrypted before written to disk
- Decrypted when read into memory
- TDE does not increase the size of the database
- TDE will impact performance of the appliance

Scope of TDE

- Control Node
- Compute Node
- Database files
- Backups
- PDW Engine log files
- Tempdb and pdwtempdb

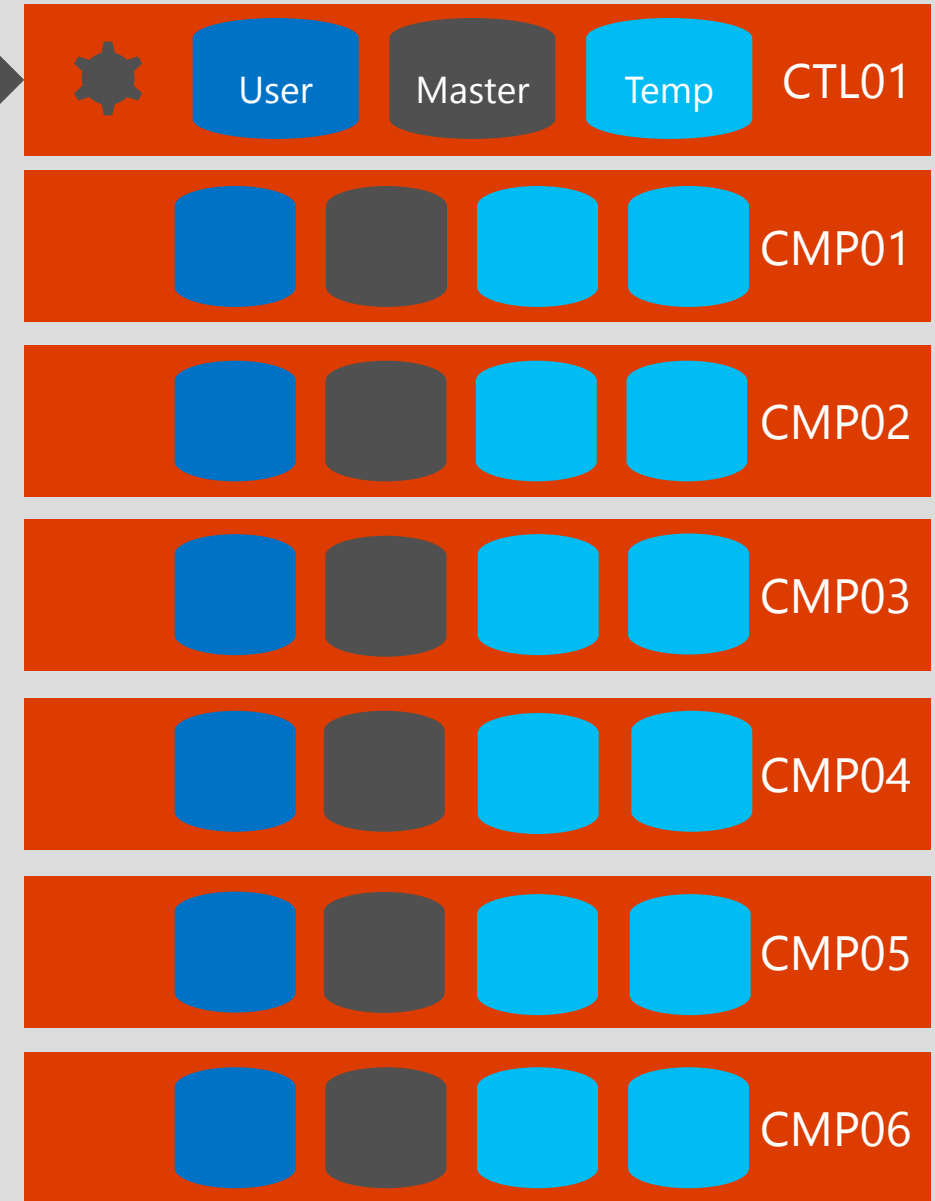
Out of Scope

- Encryption between client and PDW
- Encryption during Internal DMS operations
- Cell-level encryption (EncryptByKey/DecryptByKey)
- Shell master database isn't encrypted
- External tables aren't encrypted
- Diagnostic Sessions aren't encrypted
- Memory Dumps

How TDE Works

Initiate

1. User Creates Master Key in master
 1. PDW Creates Master Key on CTL01
 2. PDW Creates Separate Master Key (all CMP)
2. User Enables Appliance encryption
 1. PDW Encrypts tempdb & pdwtempdb
3. User Creates Certificate in master
 1. PDW Creates Certificate on CTL01
 2. PDW Exports Certificate & Imports it (all CMP)
4. User Creates Database Encryption Key
 1. PDW Creates Database Encryption key on CTL01
 2. PDW Creates Different Database Encryption Key (all CMP)
5. Initiate Database Encryption
 1. PDW Encrypts user database



Consequences & Limitations

- Both TempDB's are encrypted – impacts performance and sizing (tempdb, pdwtempdb)
- sys.certificates: user generated certificates only
- Principals need CONTROL SERVER to create certificates & Database Encryption Keys
- PDW logs by default are also encrypted which may impact support call duration

Enabling TDE in the Appliance

```
-- Create Master Key in master database:  
USE master;  
CREATE MASTER KEY ENCRYPTION  
BY PASSWORD = 'P@55w0rd';  
GO
```

Enabling TDE in the Appliance

-- Enable encryption:

```
EXEC sp_pdw_database_encryption 1;
```

```
GO
```

-- Disable encryption:

```
EXEC sp_pdw_database_encryption 0;
```

```
GO
```

-- Rotate certificate and DEK for internal databases.

This will succeed only if TDE is enabled:

```
EXEC sp_pdw_database_encryption_regenerate_system_keys;
```

```
GO
```

Enabling TDE in the Database

```
-- Step 1
-- Create certificate in master database:
USE master;
CREATE CERTIFICATE cert_20140101
    WITH SUBJECT= 'PDW DB certificate 2014-01-01';
GO
```

Enabling TDE in the Database

```
-- Step 2
-- Back up the certificate with private key
BACKUP CERTIFICATE cert_20140101
    TO FILE = '\\SECURE_SERVER\cert\20140101.cer'
    WITH PRIVATE KEY
    (
        FILE = '\\SECURE_SERVER\cert\20140101.key'
    ,   ENCRYPTION BY PASSWORD = 'P@55w0rd2'
    );
GO
```

Enabling TDE in the Database

```
-- Step 3
-- Create DEK in user DB
USE PDW_AU1_DB;
CREATE DATABASE ENCRYPTION KEY
    WITH ALGORITHM = AES_128
-- AES_192 AES_256 TRIPLE_DES_3KEY also supported...
    ENCRYPTION BY SERVER CERTIFICATE cert_20140101;
GO
```

Enabling TDE in the Database

```
-- Step 4
-- Turn the encryption on
ALTER DATABASE PDW_AU1_DB SET ENCRYPTION ON;
GO
```

PDW Engine Log Encryption

-- Disable masking - queries & data appear in PDW Engine Log

EXEC sp_pdw_log_user_data_masking 0;

GO

-- Enable masking of PDW Engine Log

EXEC sp_pdw_log_user_data_masking 1;

GO

-- Remove queries completely from PDW Engine Log

EXEC sp_pdw_log_user_data_masking 2;

GO

-- Show Current State

EXEC sp_pdw_log_user_data_masking;

GO

Catalog Views and DMVs

- `sys.databases`
- `sys.certificates`
- `sys.dm_pdw_nodes_database_encryption_keys`

PDW Permissions Model

Similar to SQL Server

Server Principals

- Logins
- Server roles

Database Principals

- Database Roles
- Users
- User defined Roles

Permissions

- Grant
- Revoke
- Deny

User Defined Roles are
recommended

Adding users to fixed
database roles requires
membership of db_owner

Exceptions

- Mixed mode authentication
 - cannot be changed to trusted only
- sa account
 - Cannot be disabled or removed
- Server Roles
 - Cannot be joined
 - Cannot be created
 - sa account is sole member of sysadmin

sa needed to add first
user to db_owner

User defined database
roles more flexible and
adhere to SQL Servers
“modern” security regime

Creating administrators in PDW

- Use CONTROL SERVER for sa level permissions

Permissions Model for ELT

Server level

- ADMINISTER BULK OPERATIONS
- VIEW SERVER STATE

Database Level

- ALTER
- VIEW DEFINITION

Databases

- Staging
- User Database

Control Server

TDE

- Certificates
- Keys
- Procedures
 - Sp_pdw_database_encryption
 - Sp_pdw_database_encryption_regenerate_system_keys
 - Sp_pdw_log_user_data_masking

Polybase

- External Data Source
- External File Format

DMVs

- nodes_columns
- nodes_indexes
- nodes_partitions

Administer Bulk Operations

Polybase

- Create External Table
 - Also Required
 - Create Table
 - Alter Schema

Data Loading

View Server State

Required for

- Management Console
- Catalog Views
 - Columnstore
 - Server permissions
- DMVs
 - Connections
 - Requests
 - Sessions

- DBCC Commands

- PDW_SHOWPARTITIONSTATS()
- PDW_SHOWEXECUTIONPLAN
- PDW_SHOWSPACEUSED

Alter Server State

DBCC Commands

- Shrinklog
- Freeproccache

Configuring Backup

- sp_pdw_add_network_credentials
- sp_pdw_remove_network_credentials

Sessions

- Kill

Diagnostic Sessions

- Create Diagnostic Sessions

Granting Access to Server State DMVs

...Without granting View Server State...

Use a Stored Procedure

- Select from sys.dm_pdw_exec_requests
- Reduce columns
- Filter rows

```

CREATE PROCEDURE dbo.logicalRequests @pSession_id nvarchar(32)
AS
SELECT
    request_id          AS Request_ID
    ,session_id         AS Session_ID
    ,[status]           AS Request_Status
    ,submit_time        AS Request_SubmitTime
    ,start_time         AS Request_StartTime
    ,end_compile_time   AS Request_EndCompileTime
    ,end_time           AS Request_EndTime
    ,total_elapsed_time AS Request_TotalElapsedDuration_ms
    ,DATEDIFF(ms,submit_time,start_time) AS Request_InitiateDuration_ms
    ,DATEDIFF(ms,start_time,end_compile_time) AS Request_CompileDuration_ms
    ,DATEDIFF(ms,end_compile_time,end_time) AS Request_ExecDuration_ms
    --,[label]           as Request_QueryLabel
    ,CASE WHEN [label] IS NOT NULL
        THEN N'<Sensitive Info Access Denied>'
        ELSE command
    END
    as Request_Command
    ,database_id        as Request_Database_ID
FROM sys.dm_pdw_exec_requests
WHERE session_id = @pSession_id;

```

Workload Management

Workload Management

- PDW engine manages a pool of 32 concurrency "slots"
- Maximum number of queries executing at the same time is 32
- Slots can also be bundled together to increase the resources allocated to a user's query
- The bundles are defined in PDW as server roles
- Resource Governor allocates resources to server roles

Why Is Workload Management important?

Enables you to

- Reduce concurrent queries running on PDW
- Allocate more resources to running queries
- Evenly affinitizes resources across appliance

Resource Classes

- Server role = Resource class
- Only non-default resource classes are visible

```
SELECT  sp1.name          AS owner_principal_name
        , sp1.type_desc AS owner_type_desc
        , sp.name         AS server_principal_name
        , sp1.type_desc AS server_type_desc
FROM    sys.server_principals sp
JOIN    sys.server_principals sp1
ON      sp.owning_principal_id
=       sp1.principal_id
WHERE   sp.type_desc = 'SERVER_ROLE'
AND     sp1.name      = 'NT SERVICE\sqlengine'
;
```

	owner_principal_name	owner_type_desc	server_principal_name	server_type_desc
1	NT SERVICE\sqlengine	WINDOWS_LOGIN	xlargerc	WINDOWS_LOGIN
2	NT SERVICE\sqlengine	WINDOWS_LOGIN	largerc	WINDOWS_LOGIN
3	NT SERVICE\sqlengine	WINDOWS_LOGIN	mediumrc	WINDOWS_LOGIN

Resource Pools

```
SELECT DISTINCT
    rp.name
    ,max_memory_kb*1.0/1048576 AS rp_max_mem_GB
    ,min_cpu_percent
    ,max_cpu_percent
    ,min_memory_percent
    ,max_memory_percent
FROM sys.dm_pdw_nodes_resource_governor_resource_pools rp
JOIN sys.dm_pdw_nodes nod ON rp.pdw_node_id = nod.pdw_node_id
WHERE rp.name <> 'internal'
AND nod.[type] = 'compute';
```


Resource Pool Configuration

	name	rp_max_mem_GB	min_cpu_percent	max_cpu_percent	min_memory_percent	max_memory_percent
1	default	22.844551086	0	100	0	100
2	QueryPool_A	20.940818786	0	100	11	11
3	QueryPool_B	20.940818786	0	100	11	11
4	QueryPool_C	20.940818786	0	100	11	11
5	QueryPool_D	20.940818786	0	100	11	11
6	QueryPool_E	20.940818786	0	100	11	11
7	QueryPool_F	20.940818786	0	100	11	11
8	QueryPool_G	20.940818786	0	100	11	11
9	QueryPool_H	20.940818786	0	100	11	11

- CPU not impacted by Resource Governor
- Pools A-H each have 11% of the memory

Workload Groups

```
SELECT DISTINCT
    rp.name                                AS rp_name
    ,wg.name                                AS wg_name
    ,importance
    , max_memory_kb*1.0/1048576            AS rp_max_mem_GB
    ,(max_memory_kb*1.0/1048576/100)
    *request_max_memory_grant_percent AS request_max_memory_grant_GB
FROM sys.dm_pdw_nodes_resource_governor_resource_pools rp
JOIN sys.dm_pdw_nodes nod ON rp.pdw_node_id = nod.pdw_node_id
JOIN sys.dm_pdw_nodes_resource_governor_workload_groups wg
    ON rp.pdw_node_id= wg.pdw_node_id
    AND rp.pool_id= wg.pool_id
WHERE rp.name <> 'internal'
AND nod.[type] = 'compute'
ORDER BY rp_name,request_max_memory_grant_GB;
```

Workload Group Configuration

	rp_name	wg_name	importance	rp_max_mem_GB	request_max_memory_grant_GB
1	default	default	Medium	22.844551086	5.711138
2	QueryPool_A	QueryGroup_A	Medium	20.940818786	0.628225
3	QueryPool_A	QueryGroupMedium_A	Medium	20.940818786	1.884674
4	QueryPool_A	QueryGroupLarge_A	High	20.940818786	4.397572
5	QueryPool_A	QueryGroupExtraLarge_A	High	20.940818786	13.820940
6	QueryPool_B	QueryGroup_B	Medium	20.940818786	0.628225
7	QueryPool_B	QueryGroupMedium_B	Medium	20.940818786	1.884674
8	QueryPool_B	QueryGroupLarge_B	High	20.940818786	4.397572
9	QueryPool_B	QueryGroupExtraLarge_B	High	20.940818786	13.820940

- Only Large and extra Large have High Priority
- Memory Grant increases significantly with larger classes
- Each Pool has separate workload group for the class

NUMA Node Mapping

```
SELECT DISTINCT
    node_affinity
    ,local_tcp_port
    ,rp.name
FROM sys.dm_pdw_nodes_exec_connections con
JOIN sys.dm_pdw_nodes_exec_sessions sess
ON    con.most_recent_session_id = sess.session_id
AND   con.pdw_node_id = sess.pdw_node_id
JOIN sys.dm_pdw_nodes nod
ON    con.pdw_node_id = nod.pdw_node_id
JOIN sys.dm_pdw_nodes_resource_governor_workload_groups wg
ON    sess.group_id = wg.group_id
AND   sess.pdw_node_id = wg.pdw_node_id
JOIN sys.dm_pdw_nodes_resource_governor_resource_pools rp
ON    rp.pool_id = wg.pool_id
AND   rp.pdw_node_id = wg.pdw_node_id
WHERE nod.[type] = 'compute'
AND    wg.name <> 'default';
```

NUMA Node Mapping Result

	node_affinity	local_tcp_port	name
1	0	1500	QueryPool_A
2	1	1501	QueryPool_B
3	2	1502	QueryPool_C
4	3	1503	QueryPool_D
5	4	1504	QueryPool_E
6	5	1505	QueryPool_F
7	6	1506	QueryPool_G
8	7	1507	QueryPool_H

- Soft NUMA configuration clearly visible
- Resource Pools

NUMA TCP Ports

```
SELECT node_affinity
,local_tcp_port
,net_transport
,protocol_type
,encrypt_option
,SUM(CASE WHEN sess.[program_name] LIKE 'RCMedium%' THEN 1 ELSE 0 END) 'RCMedium'
,SUM(CASE WHEN sess.[program_name] LIKE 'RCLarge%' THEN 1 ELSE 0 END) 'RCLarge'
,SUM(CASE WHEN sess.[program_name] LIKE 'RCXLarge%' THEN 1 ELSE 0 END) 'RCXLarge'
FROM sys.dm_pdw_nodes_exec_connections con
JOIN sys.dm_pdw_nodes_exec_sessions sess
ON con.most_recent_session_id = sess.session_id
and con.pdw_node_id = sess.pdw_node_id
JOIN sys.dm_pdw_nodes nod
ON con.pdw_node_id = nod.pdw_node_id
JOIN sys.dm_pdw_nodes_resource_governor_workload_groups wg
ON sess.group_id = wg.group_id
AND sess.pdw_node_id = wg.pdw_node_id
WHERE nod.[type] = 'compute'
AND wg.name <> 'default'
GROUP BY node_affinity
,local_tcp_port
,net_transport
,protocol_type
,encrypt_option;
```

	node_affinity	local_tcp_port	net_transport	protocol_type	encrypt_option	RCMedium	RCLarge	RCXLarge
1	0	1500	TCP	TSQL	FALSE	20	8	2
2	1	1501	TCP	TSQL	FALSE	20	8	2
3	2	1502	TCP	TSQL	FALSE	20	8	2
4	3	1503	TCP	TSQL	FALSE	20	8	2
5	4	1504	TCP	TSQL	FALSE	20	8	2
6	5	1505	TCP	TSQL	FALSE	20	8	2
7	6	1506	TCP	TSQL	FALSE	20	8	2
8	7	1507	TCP	TSQL	FALSE	20	8	2

Query Affinity

Compute Nodes use Soft NUMA for resource affinity to distribution aligned queries

- Resource Class Name is embedded in the session program name
 - Used for resource pool assignment
- 8 TCP Port Server Side listeners 1500-1507 exist on each Compute Node
 - map to distributions A-H via Resource Pools
- TCP port is used to affinitize the query
 - Port maps to correct workload group for resource allocation

Concurrency Matrix

Resource Class	Request Importance	Max Memory (Per Distribution)	Query Concurrency Slots used
Default	Medium	400MB	1
Medium	Medium	1200MB	3
Large	High	2.8GB	7
XLarge	High	8.4GB	22

Maximum
Memory figure
here is "real
world" figure

Resource Governor
figure is a theoretical
max

Remember 32 is the
maximum number of
query concurrency slots

Switching Resource Classes

```
ALTER SERVER ROLE xlargerc ADD MEMBER JRJ;  
ALTER SERVER ROLE xlargerc DROP MEMBER JRJ;
```

- Login can be a member of > 1 Resource Class
- Max Resource Class is the one that is used
- Don't be over-zealous – especially with xlarge
- Requires CONTROL SERVER or membership of the role being adjusted

Identify Resource Class

Of a request

```
SELECT
    request_id
  , session_id
  , [status]
  , command
  , resource_class
FROM sys.dm_pdw_exec_requests;
```

Of a Login

```
SELECT
    sp1.name          AS owner_principal_name
  , sp1.type_desc     AS owner_type_desc
  , sp.name           AS server_principal_name
  , sp1.type_desc     AS server_type_desc
  , sp2.name          AS member_principal_name
  , sp2.type_desc     AS member_type_desc
FROM sys.server_principals sp
JOIN sys.server_principals sp1
ON sp.owning_principal_id = sp1.principal_id
JOIN sys.server_role_members rm
ON sp.principal_id = rm.role_principal_id
JOIN sys.server_principals sp2
ON rm.member_principal_id = sp2.principal_id
WHERE sp.type_desc = 'SERVER_ROLE'
AND sp1.name = 'NT SERVICE\sqlwengine';
```

```

WITH rw
AS
(SELECT session_id
,        request_id
,        [type]
,        [object_type]
,        [object_name]
,        request_time
,        acquire_time
,        DATEDIFF(ms,request_time,acquire_time) AS acquire_duration_ms
,        concurrency_slots_used
,CASE WHEN resource_class = 'smallrc' THEN 1
      WHEN resource_class = 'mediumrc' THEN 3
      WHEN resource_class = 'largerc' THEN 7
      WHEN resource_class = 'xlargerc' THEN 22
      ELSE 0 END request_resource_class
FROM sys.dm_pdw_resource_waits
)
SELECT request_id
,        concurrency_slots_used
,        request_resource_class - concurrency_slots_used AS slots_required
FROM rw;

```

Waiting for your turn?

Backups

Backups in PDW

Simple Recovery Model

- Full
- Differential

A full backup must have occurred before a differential can take place

Backup surface area

- Shell Database
- Compute Databases
- Appliance Metadata

Backup Server Configuration

Backup Server pre-requisites

- Windows file system
- Storage to backup database
- UNC file share
- Customer domain username & password
- Ideally connectivity to Infiniband (IB) network

PDW Backup Configuration

- `sp_pdw_add_network_credentials`
- Use Infiniband IP addresses
- One entry per IP network
- `sys.dm_pdw_network_credentials`

Example Backup Syntax: Diff

```
BACKUP DATABASE [AdventureWorksPDW2012]
TO DISK =
'\\172.16.254.100\pdwbackups\aw_20140705_001'
WITH
(    DIFFERENTIAL
,    NAME          = 'AdventureWorksPDW2012'
,    DESCRIPTION   = 'AdventureWorksPDW2012 backup'
);
```

Standard Backup Process

- Initiated via T-SQL
 - Use SQL Server Agent Loading Server
- User database set to read only during backup
 - Use IB network to minimize duration
- One user database at a time
- Four PDW databases backed up in parallel
- All files written to fileshare using embedded credentials

“Backup” to Hadoop

Export data to HDFS

- Low cost storage
- Agnostic format
- Avoids backup constraints
 - Database available for write
 - Partial backup
 - Parallel backups
- Avoids restore constraints
 - No DB Restore size restriction

Interesting use cases

- Obfuscation
- Sub-setting
- Archive
- Hybrid analysis with Azure

Be aware

- No concurrency control
- No hard consistency point
- Additional load on Hadoop

Integration with Backup products

- PDW database backups do not conform with SQL SMP standard backups
- No direct integration with third party products

Agnostic Approach

- Backup database to file
- Configure a filesystem backup with backup software

Restore Database

Restore database

User Database

- Restore from backup location

Master Database

- Restore using dwconfig.exe

Restore Process

- Initiated via T-SQL
- One user database at a time
- One Backup or Restore process
- All databases restored in parallel
- Must be to same size appliance or larger
 - If larger PDW automatically re-distributes the data during restore
 - If smaller PDW will raise an exception

Restore Syntax

Full Restore

```
RESTORE DATABASE Database_Name  
FROM DISK = '\\<UNC_path>\full_backup_folder';
```

Differential Restore

```
RESTORE DATABASE <Database_Name>  
FROM DISK = '\\<UNC>_path\differential_backup_folder'  
WITH BASE = '\\<UNC_path>\full_backup_folder';
```

Header Only

```
RESTORE HEADERONLY  
FROM DISK = '\\UNC_path\backup_folder';
```


Monitoring PDW

System Center Integration

- Management packs available
 - Hadoop Region uses Ambari
- Uses SQL Authentication
- Requires Login with View Server State
- Executes Appliance Health DMV queries
- Polling Frequency 15 minutes

Rolling your own Monitoring

- APS can be configured to report via hardware
 - iLO - HP
 - iDRAC – DELL
- PDW doesn't support pro-active notifications
 - DMV Polling is required
- 3rd Party monitoring solutions: limited support
 - Generic monitoring agent modules can call SQL code

PDW Data Collection Frequency

- Cluster – 60 Seconds
- Heartbeat – 60 Seconds
- Perf counters – 3 Seconds
- All other components – 5 minutes

Alerts

Active Alerts

- `sys.dm_pdw_component_health_active_alerts`
- `sys.dm_pdw_errors`
- `sys.pdw_health_alerts`

Alert History

- `sys.dm_pdw_component_health_alerts`
- Last 10,000 alerts retained in DMVs
- Last 500 shown in Management Console

Health Alerts

SELECT

```
    pha.alert_name  
, n.name  
, n.[type]  
, hcg.group_name  
, phc.component_name  
, haa.component_instance_id  
, pha.[state]  
, pha.[status]  
, pha.[type]  
, pha.[description]  
, pha.[condition]  
, haa.create_time
```

FROM sys.pdw_health_alerts pha

```
JOIN sys.dm_pdw_component_health_alerts haa on pha.component_id = haa.component_id  
                                           and pha.alert_id      = haa.alert_id
```

```
JOIN sys.dm_pdw_nodes n on haa.pdw_node_id = n.pdw_node_id
```

```
JOIN sys.pdw_health_components phc on haa.component_id = phc.component_id
```

```
JOIN sys.pdw_health_component_groups hcg on phc.group_id = hcg.group_id
```

order by create_time desc

Active Alerts

```
select
  n.name
,n.[type]
,phc.component_name
,hcg.group_name
,cha.alert_name
,cha.severity
,cha.[state]
,cha.[description]
,haa.create_time
,haa.component_instance_id
,hcg.group_name + ' - ' + phc.component_name
,haa.current_value
,cha.[type]
,haa.previous_value
from sys.dm_pdw_component_health_active_alerts haa
JOIN sys.dm_pdw_nodes n
JOIN sys.pdw_health_components phc
JOIN sys.pdw_health_component_groups hcg
JOIN sys.pdw_health_alerts cha
on haa.pdw_node_id = n.pdw_node_id
on haa.component_id = phc.component_id
on phc.group_id = hcg.group_id
ON haa.alert_id = cha.alert_id;
```

PDW Errors

```
SELECT n.name
,e.[type]
,e.session_id
,e.request_id
,e.thread_id
,e.source
,e.create_time
,e.details
,e.spid
,e.error_id
,e.request_id
,e.session_id
FROM sys.dm_pdw_errors e
JOIN sys.dm_pdw_nodes n on e.pdw_node_id= n.pdw_node_id;
```


Components

Nodes

- `sys.dm_pdw_nodes`

Component

- `sys.pdw_health_component_groups`
- `sys.pdw_health_components`
- `sys.pdw_health_component_properties`

Health Status

- `sys.dm_pdw_component_health_status`

Component Details

SELECT

```
n.name  
,n.[type]  
,phc.component_name  
,hcp.property_name  
,hcp.physical_name  
,haa.property_value  
,haa.update_time  
,haa.component_instance_id
```

FROM sys.dm_pdw_component_health_status haa

JOIN sys.dm_pdw_nodes n

on haa.pdw_node_id = n.pdw_node_id

JOIN sys.pdw_health_components phc

on haa.component_id = phc.component_id

JOIN sys.pdw_health_component_properties hcp

on haa.property_id = hcp.property_id;

Storage Capacity

- Management Console hard coded to top 10 databases
- Use DMVs!

Storage DMVs for database file sizes

- `sys.dm_pdw_nodes_os_performance_counters`
 - Data File(s) Size%
 - Log File(s) Size%
- } Performance Counters

Storage Alerts

PDW will automatically generates alerts

- Warning <30% on any volume
- Critical <10% on any volume

