Steps for SQL Server Installation and Configuration

https://odetodata.com/2015/02/installation-and-configuration-checklist-for-microsoft-sql-server/ -- **Author Cristian Satnic**

Below is the checklist used over the years for installing and configuring Microsoft SQL Server. It is by no means complete and there are certainly scenarios where additional steps will be required. It is intended to be a starting point, enabling somebody to not only standardize SQL Server installations but also to make sure that no critical steps will be missed.

Hopefully others will find it useful as well.

Before SQL Server Installation

- install and patch the OS (minimum 100 GB for system drive)
- BIOS, firmware & driver updates
  - check power management settings
    - BIOS power management should be set to OS control or disabled
    - Windows Power Plan set to 'High Performance'
- check processor speed using Task Manager and CPU-Z utility (make sure CPU runs at max speed)
- RAID 1 for OS, logs and backups (but each on a separate volume), RAID 10 for data files
- format drives for data & logs to use 64 KB allocation unit size (ideal for SQL Server) – full format, not quick
- test drive performance with CrystalDiskMark & SQLIO / save screenshots/results for future reference
- configure anti-virus scan software (if any) to not scan any database related files such as MDF, NDF, LDF, BAK or TRN
- create regular user/domain accounts for each SQL Server component – main SQL service, agent, reporting, analysis, integration (SQLServiceAccount, SQLAgentAccount and so on)
  - for a dev server: probably ok to use just one SQL user account for all SQL services
  - use gpedit.msc to grant appropriate rights to the user for SQL Server Service (Computer Configuration -> Windows Settings -> Security Settings -> Local Policies -> User Rights Management)
    - Perform volume maintenance tasks
    - Lock pages in memory
      - if this is done then set sp_configure 'max server memory' to make sure the OS is left with enough memory to function
      - might want to not enable this on a dev server with shared duties (such as a webserver) so we don't end up in the scenario where SQL Server will starve out the other processes for memory
- prepare directories (on separate drives): SQLData, SQLLogs, SQLTempDB, SQLBackups
  - folder permissions
    - the install process will properly assign folder permissions to custom folders if they're all specified initially in the SQL data directories configuration
      - starting with SQL Server 2012 permissions are assigned to the per-service SID for each of its services – https://msdn.microsoft.com/en-us/library/jj219062.aspx
    - remove "Everyone" (if set) from non-C drives where SQL-related folders will exist
    - for SQL Server main service user account: give full permissions to folder where SQL will be installed and the SQLData, SQLLogs, SQLTempDB, SQLBackups folders
    - for SQL Server agent user account: give full permissions to folder where SQL will be installed and SQLBackups.

During SQL Server Installation

- only install features that are really needed (especially on a production system) – don't install something "just in case it might be needed it later"
- use the proper user accounts (created above) for each service
  - set SQL Server Agent to auto start
  - most likely disable the SQL Server Browser service
- on the Database Engine Configuration tab be sure to set the Data Directories to the custom ones we plan to use (not the defaults where everything is stored on the C drive)

After SQL Server Installation

- Always use the SQL Server Configuration Manager tool to make changes to SQL services (don't do it from the Control Panel)
- Install all the latest SQL Server service packs & cumulative updates – http://sqlserverbuilds.blogspot.com/
- SQL Server properties in SSMS (make changes by scripting them from SSMS to confirm exactly what we're about to change)
  - Database Settings – enable backup compression
  - Advanced – enable "Optimize for Ad hoc Workloads"
  - look over the other settings as well
- Enable DAC (remote dedicated administrator connection) – https://msdn.microsoft.com/en-us/library/ms190468.aspx
- Make sure that in SQL Server Configuration Manager we have TCP/IP enabled
  - allow SQL ports to receive incoming connections at the OS firewall
- From a client computer: create a test data link file (file with *.udl extension) and set its connection properties to test if that particular client can reach the new SQL Server instance
- Create TempDB files to correspond to 1/4 – 1/2 the number of processor cores in the system (for each file – 4096 MB initial size, 1024 MB autogrowth / for log – 1024 MB initial size, 512 MB autogrowth)
- Enable Database Mail
  - enable Database Mail XPs – https://msdn.microsoft.com/en-us/library/ms191189.aspx
  - need to have info for an outgoing SMTP server
  - create Database Mail account/profile – call it SQLDBAGroup
- Set traceflag in SQL 2012/2014 to deal with jumps in identity values after system restarts
  - http://www.codeproject.com/Tips/668042/SQL-Server-Auto-Identity-Column-Value-Jump-Is
  - https://connect.microsoft.com/SQLServer/feedback/details/739013/failover-or-restart-results-in-reseed-of-identity
- Set up SQL dashboard reports for system maintenance & diagnostics (links below work for both SQL Server 2012 and 2014)
  - http://blogs.msdn.com/b/sql_pfe_blog/archive/2013/04/15/sql-2012-system-health-reporting-dashboard-visualizing-sp-server-diagnostics-results.aspx
  - http://www.mssqltips.com/sqlservertip/2670/install-sql-server-2012-performance-dashboard-reports/
- Create SQL Server Agent operator and alerts – look at SQL Agent properties and make sure it's configured to properly send job alert emails via Database Mail
  - http://www.brentozar.com/blitz/configure-sql-server-alerts/
  - http://www.sqlskills.com/blogs/glenn/how-to-create-sql-server-agent-alerts-for-critical-errors/

- Install Ola Hallengren's Maintenance Solution – http://ola.hallengren.com/
    - create schedules for each of the SQL Agent jobs created by the maintenance script & add notification alerts to each job
    - test jobs for backups, integrity checks, index maintenance
- Install sp_Blitz, sp_BlitzIndex and other such stored procedures from Brent Ozar
    - execute them and correct any issues found
- Install sp_WhoIsActive by Adam Machanic
- check configuration with Microsoft SQL Server 2012 Best Practice Analyzer – http://www.mssqltips.com/sqlservertip/2712/using-the-microsoft-sql-server-2012-best-practice-analyzer/
- Security best practices
    - these can be lists/articles all by themselves, so I'll just point to some of the best ones out there
    - http://www.greensql.com/content/sql-server-security-best-practices
    - http://www.mssqltips.com/sql-server-tip-category/19/security/