```powershell
Add-PSSnapin Microsoft.SharePoint.PowerShell -erroraction SilentlyContinue

## Settings you may want to change ##
$databaseServerName = "DB001.DOMAIN.COM"
$searchServerName = $databaseServerName
$saAppPoolName = "SharePoint Web Services"
$appPoolUserName = "DOMAIN\SP_POOL"


## Service Application Service Names ##

$bcsSAName = "Business Data Connectivity Service"
$metadataSAName = "Managed Metadata Web Service"
$stateSAName = "State Service"
$secureStoreSAName = "Secure Store Service"
$usageSAName = "Usage and Health Data Collection Service"
$userProfileSAName = "User Profile Synchronization Service"
$WordAutomationSAName = "Word Automation Services"



$saAppPool = Get-SPServiceApplicationPool -Identity $saAppPoolName -EA 0
if($saAppPool -eq $null)
{
  Write-Host "Creating Service Application Pool..."

  $appPoolAccount = Get-SPManagedAccount -Identity $appPoolUserName -EA 0
  if($appPoolAccount -eq $null)
  {
      Write-Host "Please supply the password for the Service Account..."
      $appPoolCred = Get-Credential $appPoolUserName
      $appPoolAccount = New-SPManagedAccount -Credential $appPoolCred -EA 0
  }

  $appPoolAccount = Get-SPManagedAccount -Identity $appPoolUserName -EA 0

  if($appPoolAccount -eq $null)
  {
    Write-Host "Cannot create or find the managed account $appPoolUserName, please
ensure the account exists."
    Exit -1
  }

  New-SPServiceApplicationPool -Name $saAppPoolName -Account $appPoolAccount -EA 0 >
$null

}


Write-Host "Creating BCS Service and Proxy..."
New-SPBusinessDataCatalogServiceApplication -Name $bcsSAName -ApplicationPool
$saAppPoolName -DatabaseServer $databaseServerName -DatabaseName
"BusinessDataCatalogDB" > $null
Get-SPServiceInstance | where-object {$_.TypeName -eq "Business Data Connectivity
Service"} | Start-SPServiceInstance > $null


Write-Host "Creating Metadata Service and Proxy..."
New-SPMetadataServiceApplication -Name $metadataSAName -ApplicationPool $saAppPoolName
-DatabaseServer $databaseServerName -DatabaseName "MetadataDB" > $null
New-SPMetadataServiceApplicationProxy -Name "$metadataSAName Proxy" -DefaultProxyGroup
-ServiceApplication $metadataSAName > $null
Get-SPServiceInstance | where-object {$_.TypeName -eq "Managed Metadata Web Service"}
| Start-SPServiceInstance > $null


Write-Host "Creating State Service and Proxy..."
New-SPStateServiceDatabase -Name "TS_StateServiceDB_2013" -DatabaseServer
$databaseServerName | New-SPStateServiceApplication -Name $stateSAName | New-
SPStateServiceApplicationProxy -Name "$stateSAName Proxy" -DefaultProxyGroup > $null
```

```
Write-Host "Creating Secure Store Service and Proxy..."
New-SPSecureStoreServiceapplication -Name $secureStoreSAName -Sharing:$false -
DatabaseServer $databaseServerName -DatabaseName "SecureStoreServiceAppDB" -
ApplicationPool $saAppPoolName -auditingEnabled:$true -auditlogmaxsize 30 | New-
SPSecureStoreServiceApplicationProxy -name "$secureStoreSAName Proxy" -
DefaultProxygroup > $null
Get-SPServiceInstance | where-object {$_.TypeName -eq "Secure Store Service"} | Start-
SPServiceInstance > $null



Write-Host "Creating User Profile Service and Proxy..."
$userProfileService = New-SPProfileServiceApplication -Name $userProfileSAName -
ApplicationPool $saAppPoolName -ProfileDBServer $databaseServerName -ProfileDBName
"TS_ProfileDB_2013" -SocialDBServer $databaseServerName -SocialDBName
"TS_SocialDB_2013" -ProfileSyncDBServer $databaseServerName -ProfileSyncDBName
"TS_SyncDB_2013"
New-SPProfileServiceApplicationProxy -Name "$userProfileSAName Proxy" -
ServiceApplication $userProfileService -DefaultProxyGroup > $null
Get-SPServiceInstance | where-object {$_.TypeName -eq "User Profile Service"} | Start-
SPServiceInstance > $null




Write-Host "Creating Word Conversion Service and Proxy..."
New-SPWordConversionServiceApplication -Name $WordAutomationSAName -ApplicationPool
$saAppPoolName -DatabaseServer $databaseServerName -DatabaseName "WordAutomationDB" -
Default > $null
Get-SPServiceInstance | where-object {$_.TypeName -eq "Word Automation Services"} |
Start-SPServiceInstance > $null



Write-Host "Creating Search Service and Proxy..."

Add-PSSnapin Microsoft.SharePoint.PowerShell -erroraction SilentlyContinue

$searchServerName = (Get-ChildItem env:computername).value

$serviceAppName = "Search Service Application"
$searchDBName = "SearchService_DB"

$saAppPool = Get-SPServiceApplicationPool $saAppPoolName

Write-Host "Starting Search Service Instances..."
Start-SPEnterpriseSearchServiceInstance $searchServerName
Start-SPEnterpriseSearchQueryAndSiteSettingsServiceInstance $searchServerName


# Create the Search Service Application and Proxy
Write-Host "Creating Search Service Application and Proxy..."
$searchServiceApp = New-SPEnterpriseSearchServiceApplication -Name $serviceAppName -
ApplicationPool $saAppPoolName -DatabaseName $searchDBName
$searchProxy = New-SPEnterpriseSearchServiceApplicationProxy -Name "$serviceAppName
Proxy" -SearchApplication $searchServiceApp


# Clone the default Topology (which is empty) and create a new one and then activate
it

Write-Host "Configuring Search Component Topology..."
$clone = $searchServiceApp.ActiveTopology.Clone()
$searchServiceInstance = Get-SPEnterpriseSearchServiceInstance $searchServerName
New-SPEnterpriseSearchAdminComponent -SearchTopology $clone -SearchServiceInstance
$searchServiceInstance
```

```
New-SPEnterpriseSearchContentProcessingComponent –SearchTopology $clone -
SearchServiceInstance $searchServiceInstance
New-SPEnterpriseSearchAnalyticsProcessingComponent –SearchTopology $clone -
SearchServiceInstance $searchServiceInstance
New-SPEnterpriseSearchCrawlComponent –SearchTopology $clone -SearchServiceInstance
$searchServiceInstance
New-SPEnterpriseSearchIndexComponent –SearchTopology $clone -SearchServiceInstance
$searchServiceInstance
New-SPEnterpriseSearchQueryProcessingComponent –SearchTopology $clone -
SearchServiceInstance $searchServiceInstance
$clone.Activate()


Write-Host "Search Configuration Done Successfully!"



Write-Host "change the default content crawl account. By default Search uses the Farm
account, which is a really bad idea."

Read-Host -Prompt "Press Enter to exit"
```