<http://softwarehard.blogspot.in/2012/02/best-practices-for-running-aspnet-on.html> 

**When should application pools be turned into web gardens?**

Web gardens should only be used if the application doesn’t use in-process session variables but rather out-of-process ones (e.g. session state service or database session state). Reason is that a web garden would have at least 2 worker processes which do not share in-process (session) memory.   
Drivers to using web gardens are:

1. Application makes long-running synchronous requests
2. Application is low in availability and crashes often
3. Application creates high CPU load on worker process

**Best Practices Systems Settings**

1. Optimum paging file size setting:
   1. 1.5x the RAM for 32-bit OS
   2. system-managed for 64-bit OS
2. Disk queue length should always average less than 2
3. Processor queue length should be less than the number of processors
4. Network utilisation should be less than 50%

**Application Isolation Policy**

1. Some applications should be deployed into their own application pool
   1. mission critical and should be highly available
   2. needs to be isolated for troubleshooting (e.g leaks memory)
   3. expected to incur high throughput/ load. putting undue stress on server resources
   4. has non-standard requirements (e.g. configuration)

**Application Pool Settings**

1. *Enable* 32-bit Applications (WOW64) where necessary (e.g. when application uses 32-bit libraries)
2. Always *use* Managed Pipeline Mode unless application runs into issues otherwise
3. *Increase* Queue Length if required (especially if server hits error 503 – Service Unavailable)
4. Run using *Network Service* as the Identity
5. For HA requirement, *disable* Idle Timeout of the work process
6. *Disable* Rapid Fail Protection for PRD environment
7. For generating event log under the pool recycling settings, set **all** events to *true*
8. Set to *0* minutes the recycling of the application pool at Regular Time Interval

**IIS Configuration Settings**

1. Enable at least static compression
2. Turn off directory browsing

**Handling Session States**

1. Use **InProc** in standalone mode (where web farms are not used). This performs the best.
2. Use **StateServer** (perferably on a different machine) to share session state in web farms. However, the performance penalty is 15% slower than InProc. A more serious consequence is that the StateServer would now be a single point of failure.
3. Use **SQLServer** to share session state in web farms introduces 25% penalty over InProc. However, if set-up as a failover cluster, it will work reliably with HA.
4. Use **CustomProvider** for other in-memory, distributed and equally reliable session replication. These tend to perform faster than SQLServer and almost equally fast as StateServer. Some popular providers are: Windows Server AppFabric, ScaleOut SessionServer, and memcached.

**Handling MachineKey**

1. The machineKey is used to encrypt and decrypt the following: ViewState, Forms authentication, out-of-process Session data.
2. By default, the encryption and decryption keys are auto-generated by [ASP.NET](http://asp.net/) whenever the application pool restarts.
3. For web farms, do ensure that the machineKey for all the servers in the web farm are configured to use the same values (esp. validationKey, decryptionKey). Auto-generation or inconsistent values would cause unnecessary issues.