Windows performance

This document offers some performance optimizations you might like to apply to your Windows hosts to speed them up specifically in the context of using Ansible with them, and generally.

Optimize PowerShell performance to reduce Ansible task overhead

To speed up the startup of PowerShell by around 10x, run the following PowerShell snippet in an Administrator session. Expect it to take tens of seconds.

> **Note**
>
> If native images have already been created by the ngen task or service, you will observe no difference in performance (but this snippet will at that point execute faster than otherwise).

```powershell
function Optimize-PowershellAssemblies {
  # NGEN powershell assembly, improves startup time of powershell by 10x
  $old_path = $env:path
  try {
    $env:path = [Runtime.InteropServices.RuntimeEnvironment]::GetRuntimeDirectory()
    [AppDomain]::CurrentDomain.GetAssemblies() | % {
      if (! $_.location) {continue}
      $Name = Split-Path $_.location -leaf
      if ($Name.startswith("Microsoft.PowerShell.")) {
        Write-Progress -Activity "Native Image Installation" -Status "$name"
        ngen install $_.location | % {"`t$_"}
      }
    }
  } finally {
    $env:path = $old_path
  }
}
Optimize-PowershellAssemblies
```

PowerShell is used by every Windows Ansible module. This optimization reduces the time PowerShell takes to start up, removing that overhead from every invocation.

This snippet uses the native image generator, ngen to pre-emptively create native images for the assemblies that PowerShell relies on.

Fix high-CPU-on-boot for VMs/cloud instances

If you are creating golden images to spawn instances from, you can avoid a disruptive high CPU task near startup via processing the ngen queue within your golden image creation, if you know the CPU types won't change between golden image build process and runtime.

Place the following near the end of your playbook, bearing in mind the factors that can cause native images to be invalidated (see MSDN).

```yaml
- name: generate native .NET images for CPU
  win_dotnet_ngen:
```