

Eze Castle Integration

## **ECI VM Automation Architecture**

Design Version: 1.0 Date: 6-12-2018

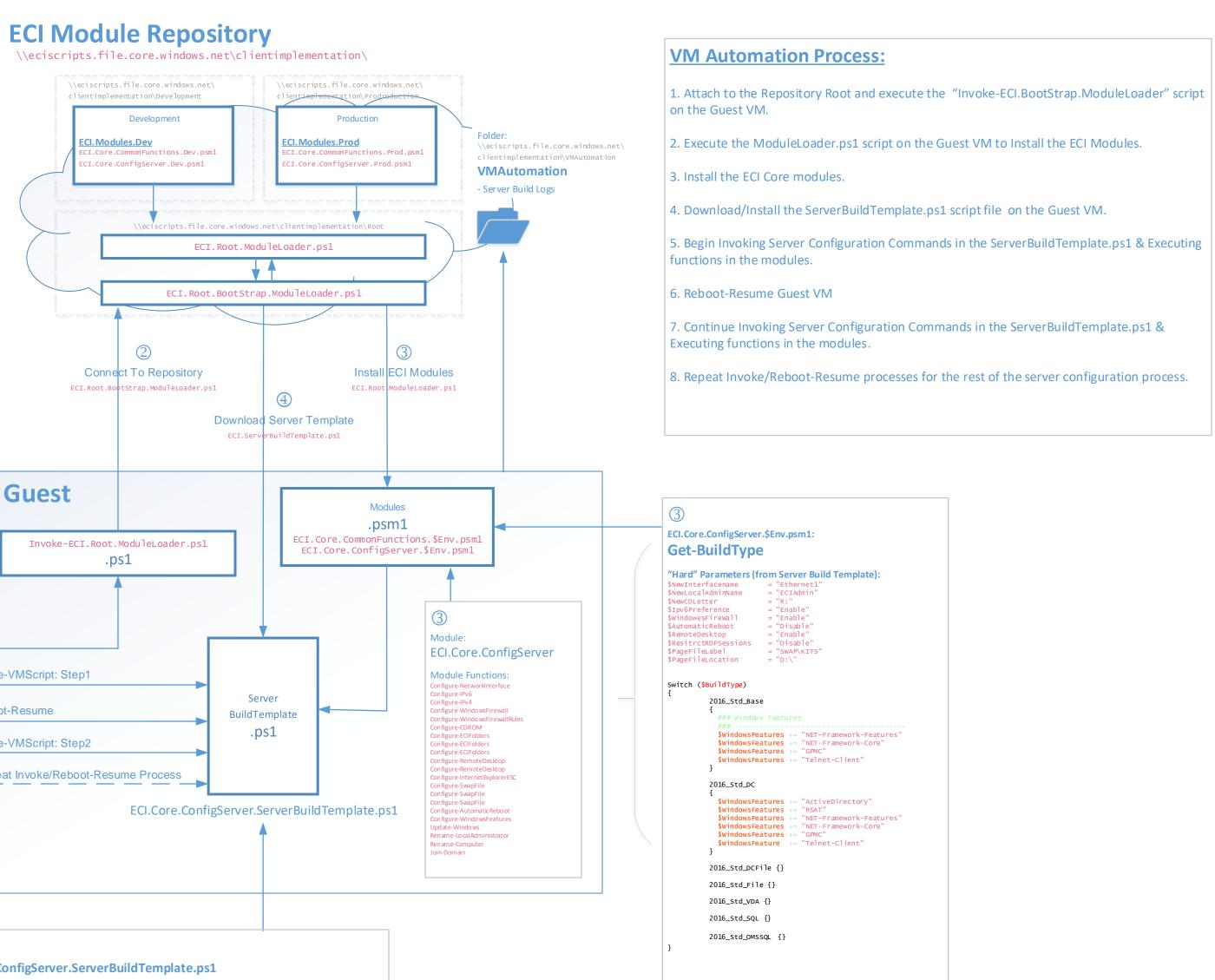
Write-Host "Executing ConfigServer Functions: -Step: " \$Step

Write-Host "Executing ConfigServer Functions: -Step: " \$Step

Write-Host "Executing ConfigServer Functions: -Step: "\$Step
Rename-LocalAdministrator #<-- Need code to Re-Authenticate with new ECIAdmin Credentials

if (\$Step -eq "6")

if (\$Step -eq "7")



```
ECI.Root.ModuleLoader.ps1
 ### Set Repository Environments
  function Set-ECIEnvironment
    [Alias("Set-ECI")]
[Alias("Set-Env")]
[Alias("SetEnv")]
[ValidateSet("Dev","Prod")]
    Param(
[Parameter(Mandatory = $False)] [string]$Env = "Dev")
    ### Set Repository Name Space
    ### Set Repository Variable
       $global:Env = $Env.ToUpper()

$global:Environment = $Environment.ToUpper()

$global:NameSpace = "ECI.Repository.$Environment"
         $global:RootDrive
$global:RootPath
$global:EnvDrive
$global:EnvPath
                                        = "\\eciscripts.file.core.windows.net\clientimplementation"
        ### Create Repository Object
   = $NameSpace
= $RootDrive
= $RootPath
= $EnvDrive
= $EnvPath
= $ModulePath
     $ECI = New-Object PSObject -Property $ECI
     ### Write Console Header
  ###-
Write-Host ('*' * 15)($Eci.Environment).ToUpper()('*' * 15) -Foregroundcolor Cyan
Write-Host ('-' * 100) -Foregroundcolor Cyan
Write-Host "Environment : " $ECI.Environment -ForegroundColor Cyan
Write-Host "RootDrive/Path : " $ECI.Environment -ForegroundColor Cyan
Write-Host "EnvDrive/Path : " $ECI.EnvDrive "=" $ECI.EnvPath -ForegroundColor Cyan
Write-Host "ModulePath : " $ECI.EnvDrive "=" $ECI.EnvPath -ForegroundColor Cyan
Write-Host ('-' * 100) -ForegroundColor Cyan
     ### Clear PSModulePath
    ##$CleanPath = ($env:PSModulePath.Split(";") | where-Object { $_ -notlike "*Eci.Modules*"}) -join ';'
$CleanPath = "C:\Users\$env:UserName\Documents\windowsPowerShell\Modules;C:\Program Files\WindowsPowerShell\Modules;C:\windows\system32\WindowsPowerShell\v1.0\Modules\"
$env:PSModulePath = $ECI.ModulePath + ";" + $CleanPath
     ### Remove Environmental Drives
      PSEnvDrives = Get-PSDrive -PSProvider FileSystem | Where-Object {($_.DisplayRoot -like "*eciscripts*")}
         foreach($PSEnvDrive in $PSEnvDrives)
             #write-Host "Removing Drive: $PSEnvDrive" -ForegroundColor Yellow
Remove-PSDrive -Name $PSEnvDrive -PSProvider FileSystem -Force
     ### Map Environmental Drives
     Write-Host "Mapping Root Drive: $RootDrive $RootPath" -ForegroundColor Yellow
$global:PSRootDrive = New-PSDrive -Name $ECI.RootDrive -PSProvider FileSystem -Root $ECI.RootPath -Credential $Credentials -Scope global -ErrorAction SilentlyContinue -Persist | Out-Null
    Write-Host "Mapping Environment Drive: EnvDrive $EnvPath" -ForegroundColor Yellow
$global:PSEnvDrive = New-PSDrive -Name $ECI.EnvDrive -PSProvider FileSystem -Root $ECI.EnvPath -Credential $Credentials -Scope global -Persist -ErrorAction SilentlyContinue | Out-Null
     ### Unload Previously Loaded modules
     .
Write-Host "Unloading ECI Module: " $LoadedECIModule -ForegroundColor Yellow
Remove-Module -Name $LoadedECIModule.Name
     ### Import ECI Core Modules
     """

SECIModules = Get-Module -ListAvailable -Name ECI.Core*
foreach ($ECIModule in $ECIModules)
         Write-Host "Importing Module: $ECIModule" -ForegroundColor Green
          Import-Module -Name $ECIModule -DisableNameChecking -Global -Force #-Prefix ECI
     ### Set Propmpt & Location
     """
Set-Location ($EnvDrive + ":\")
function global:Prompt {"[ECI-$Env] $(Get-Location) > "}
 Set-ECIEnvironment $ENV
```

Eze Castle Integration
ECI VM Automation Architecture
Author: Chris Brennan
Design Version: 1.0
Date: 6-12-2018

0123456789 **0123456789**