|  |
| --- |
| NSCC |
| OSYS1200 |
| Final Project |

|  |  |
| --- | --- |
| Caldwell,Simon  12-2-2024 |  |

Contents

[Introduction 2](#_Toc184763239)

[Part 1 – Vagrant Basics 3](#_Toc184763240)

[Part 2 – CompanyInc 4](#_Toc184763241)

[Part 3 – Demo App 7](#_Toc184763242)

[Part 4 – Reflection 9](#_Toc184763243)

# Introduction

In this assignment I will showcase the essentials of vagrant, displaying its capabilities and practical applications. This assignment will show the understanding of how to create, configure and provision VM’s. Including the installation and setup, then the configuration of CompanyInc by setting up directory structures, users and groups. We will also explore the creation of a new VM for a demo application. Finally, I have wrote a reflection on what I have learned over this semester.

# Part 1 – Vagrant Basics

**Below are screenshots showcasing my VM running in PowerShell and in VMWare showing my completion of part 1.**

**My VM running in VMWare:**

**A screenshot of a computer

Description automatically generated**

**My VM running in Powershell:**

**A screenshot of a computer

Description automatically generated**

# Part 2 – CompanyInc

**Below is a screenshot displaying my CompanyInc\_test.ps1 and my Vagrant file and scripts.**

**My CompanyInc\_test.ps1 running successfully:**

A screenshot of a computer

Description automatically generated

**My CompanyInc\_test.ps1 script:**

#Script Title: CompanyInc\_test.ps1

#Author: Simon Caldwell

#Description: Creation of a script to validate the directory structure and group memberships

Get-ChildItem -Path "C:\CompanyInc" -Recurse

Get-LocalGroupMember -Group "Management"

Get-LocalGroupMember -Group "Marketing"

Get-LocalGroupMember -Group "Sales"

**My NewObject.ps1 script:**

#Script Title: NewObject.ps1

#Author: Simon Caldwell

#Description: This script creates the CompanyInc directory structure, groups and users in the VM

#Create Directories

New-Item -Path "C:\CompanyInc" -ItemType Directory

New-Item -Path "C:\CompanyInc\Management" -ItemType Directory

New-Item -Path "C:\CompanyInc\Management\ABruceFiles" -ItemType Directory

New-Item -Path "C:\CompanyInc\Marketing" -ItemType Directory

New-Item -Path "C:\CompanyInc\Sales" -ItemType Directory

#Group Creation

New-LocalGroup -Name "Management"

New-LocalGroup -Name "Marketing"

New-LocalGroup -Name "Sales"

#User Creation and adding to groups

New-LocalUser -Name "ABruce" -Password (ConvertTo-SecureString -AsPlainText "School2024" -Force) -AccountNeverExpires

Add-LocalGroupMember -Group "Management" -Member "ABruce"

New-LocalUser -Name "SCaldwell" -Password (ConvertTo-SecureString -AsPlainText "Passw0rd1" -Force) -AccountNeverExpires

Add-LocalGroupMember -Group "Administrators" -Member "SCaldwell"

Add-LocalGroupMember -Group "Users" -Member "SCaldwell"

**My Vagrant File converted to text:**

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure

# configures the configuration version (we support older styles for

# backwards compatibility). Please don't change it unless you know what

# you're doing.

  Vagrant.configure("2") do |config|

    # Specify the base box

  config.vm.box = "gusztavvargadr/windows-10"

  # Provider-specific configuration for VMware

  config.vm.provider "vmware\_desktop" do |vmw|

    vmw.gui = true

     vmw.vmx["numvcpus"] = "1"

    vmw.vmx["suspend.disabled"] = "TRUE"

    vmw.vmx["memsize"] = "8192"  # Memory set to 8GB

    vmw.vmx["prefvmx.minVmMemPct"] = "100"

    vmw.vmx["mainMem.useNamedFile"] = "FALSE"

    vmw.vmx["mainMem.partialLazySave"] = "FALSE"

    vmw.vmx["mainMem.partialLazyRestore"] = "FALSE"

    end

    # Adding the provisioning script

    config.vm.provision "shell", path: "C:/Users/Simon/OneDrive/Desktop/OSYS1200/Vagrant.directory/NewObject.ps1", privileged: true

    # Sync folder to ensure the test script is available on the VM

    config.vm.synced\_folder "C:/Users/Simon/OneDrive/Desktop/OSYS1200/Vagrant.directory", "/test"

  end

# Part 3 – Demo App

**Requirements:**

The requirements for my demo app are to install Chocolatey and Node.js.

**Below is a screenshot showcasing my second VM running the application helloWorld.js:**

**A screenshot of a computer

Description automatically generated**

**Below is my helloWorld.js script:**

console.log("Hello, World!");

**Below is my Vagrantfile:**

Vagrant.configure("2") do |config|

  config.vm.box = "gusztavvargadr/windows-10"

  config.vm.provider "vmware\_desktop" do |vmw|

    vmw.gui = true

    vmw.vmx["numvcpus"] = "1"

    vmw.vmx["suspend.disabled"] = "TRUE"

  end

  # Increase boot timeout

  config.vm.boot\_timeout = 600

  # Install Chocolatey and Node.js

  config.vm.provision "shell", inline: <<-SHELL

    Set-ExecutionPolicy Bypass -Scope Process -Force; `

    [System.Net.ServicePointManager]::SecurityProtocol = `

        [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; `

    iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

    choco install nodejs -y

  SHELL

  # Synced folder for the Hello World script

  config.vm.synced\_folder "D:\DemoAppVM", "/vagrant"

end

# Part 4 – Reflection

I have learned quite a lot since the beginning of the semester of OSYS1200, such as how to create a virtual machine, how to add snap ins and check your computers diagnostics and how to write commands in PowerShell. These will be important for anytime I need to set up a new computer of my own or with the knowledge I have now help other people with windows OS. I hope to keep building on my knowledge of operating systems until I am very proficient. The biggest thing I have learned is how to be patient while using a computer and using problem solving to get a task done. I think the things I have learned in this class will help me figure things out much quicker. Thank you, Carson.