Automation and Security in Healthcare Application

Submitted by

Parth Shingala(18MCEI13)

Department of Computer Science and Engineering

Institute of Technology

Nirma University

Ahmedabad –382481

May-2020

Major Project

Report

Automation and Security in Healthcare Application

Submitted in partial fulfillment of

the requirements for the award of the degree of

Master of Technology

in

Computer Science and Engineering(Information and Networking Security)

Submitted by

Parth Shingala(18MCEI13)

Under the guidance of

Prof. Parita Oza(Assistant Professor,Nirma University,Ahmedabad)

and

Manjunath Hunsur Adishesha (Manager-DevOps,Philips innovation Campus,Bangalore)

Department of Computer Science and Engineering

Ahmedabad –382481

May-2020

Certificate

This is to certify that the Major Project entitled **Automation and Security in Healthcare Application** submitted by Parth Shingala (18MCEI13), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer science and Engineering(Information and Networking Security) of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Major Project Part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Parita Oza,
Guide and Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University,Ahmedabad

Dr. Sharada Valiveti,
Associate Professor,
Coordinator M.Tech CSE(INS),
Institute of Technology,
Nirma University,Ahmedabad

Dr. Madhuri Bhavsar,
Professor and head,
CSE Department,
Institute of Technology,
Nirma University,Ahmedabad

Dr. Alka MAhajan,
Director,
Nirma University,Ahmedabad

Statement of Originality

I, **Parth Shingala**, **18MCEI13**, give undertaking that the Major Project entitled "**Automation and Security in Healthcare Application**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering (Information and networking Security) of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student:
Date:
Place:

Endorsed by
Prof.Parita Oza
(Signature of Guide)

Acknowledgments

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof.Parita Oza**, Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Parth Shingala
18MCEI13

Abstract

As a part of Dev-Ops team,Each and DevOp need to focus on Developing the culture to bridge the gap between Developer and Operations team.Basically, DevOps is a set of practices to automate and integrate the processes between Developers and IT teams.So after development step the steps like Automation of building of source code,Maintaining and analyzing product security,Maintainig and developing the inter product automation pipeline,Automated deployment of the product , Running the automated test cases and Release of the product for verification comes under DevOps.To Address the above requests basically we are using Microsoft Team Foundation Server,Windows Powershell and Microsoft Build(msbuild).

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Project Details

A famous healthcare application of **Phillips Pvt. Ltd.** called **IntelliSpace Portal** on which my team is working on.After its development stage all the tasks like Building of source code,Implementing and analyzing the product security,Automated deployment, Running automated test cases and Releasing product for verification are being handled by DevOps team.
Being a DevOps guy tasks are given to perform are as below:

- Mail Automation[1][2]
- Developer Deploy tool[2][3]
- Maintaining CI/CD Pipeline [1]
- Nagios monitoring tool setup [5]
- STIG(Security Technical Implementation Guide) Implementation on product [4]
- Nessus scan automated pipeline creation[6]
- OVA automation implementation and pipeline creation for the same[2][8]
- TICS Pipeline creation [1][7]
- Automated build pipeline creation for 3 Applications[1][3]

1.2 Objective of work

Objective of every task which is bring assigned and DevOps is handling is to mainly automate activity which is being performed manually by team member or group of team members and maintain that automation via TFS Dashboard.

1.3 Scope of work

The above tasks can be used in any other pipelines being created later on and can be modified as per the requirements later on as per the coming request from the product management.

Chapter 2

Literature Survey

Following tools and technologies are kept in considerations while implementing relevant task:

2.1 Microsoft Team Foundation server

- TFS[1] is used for source control and version control.
- Its is used for complete lifecycle development.We can use it with different IDE like Visual Studio,Eclipse to manage the source code.
- It is combination of version control,an issue tracker like jira and a limited application lifecycle management.
- it provides services as below:
  - Souce control
  - Collaborate
  - Planning
  - Testing
  - Builds
- **Phases of CI/CD Pipeline**:
  - **Continuous Integration**:-

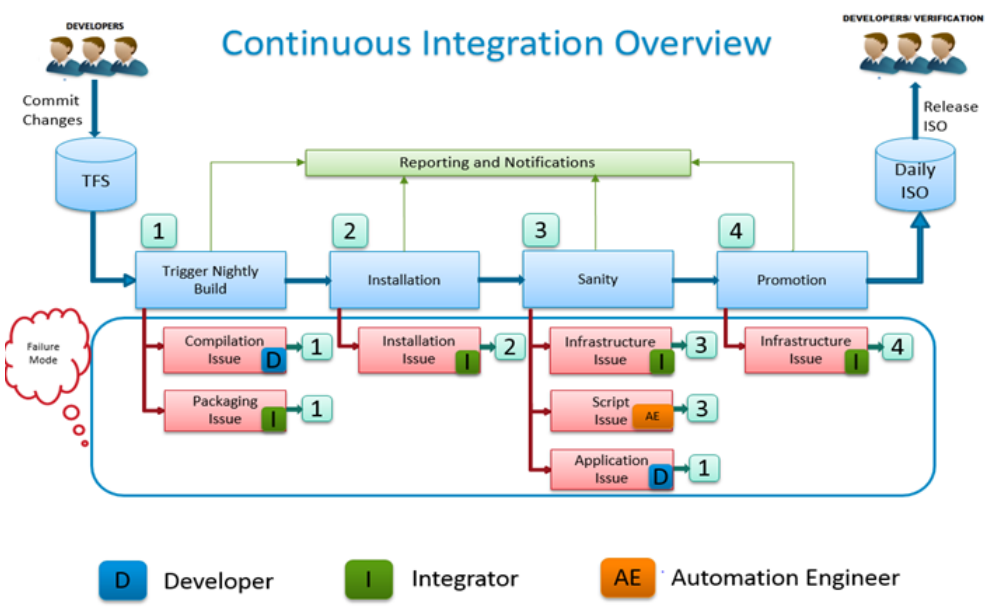


Figure 2.1: Working of TFS daily CI/CD

- ∗ This Principle came from agile culture in DevOps .
- ∗ In this Principle Developers are advised to merge their daily work into repository.This exposes integration and conflicts issue which are being faced in waterfall model earlier.

– **Continuous Testing**:-

- ∗ It creates a System of Decision that helps team to get possible business risks associated with the product.
- ∗ It guides business team to meet business requirements.
- ∗ It eliminates testing bottlenecks and simplifies the creation of virtualized test environments that can be easily modified as systems change
- ∗ After every proper implementation of Continuous Delivery developers will have a build artifacts ready ,which are passed through the testing process.
- ∗ This will result in reducing the maintenance cost of test environments.This capabilities will allow integration testing in earlier phase of life cycle,thus will result in shorting the test cycle.

– **Continuous Delivery**:-

- ∗ It can be defined as a Software practice in which the code changes ar automatically built,Deployed,Tested and prepared for release .

        * Release frequency of product differs per every company as per their policies
        * Some companies might release product every day or every week or every month.

2.2 Windows Powershell

- Windows PowerShell[2] is object-oriented scripting language and an automation engine.It is basically designed for administrators of the system . It helps software professionals to control & automate the administration of the Window operating system and applications.
- It is having combined feature of scripting flexibility,command-line speed and GUI based tool.It allows to solve problem efficiently by applying automation.It eliminates manual labour hours.
- Features of powershell
    - Powershell Remoting
    - Background Jobs
    - Transactions
    - Network File Transfer
- Reasons for using powershell
    - It is having a very good command-line experience for the operation system.
    - Allows complete access to all of the types in the .NET framework.
    - Trusted by system administrators.
    - Simple way to manipulate server and workstation components
    - Easy syntax
    - More secure than running VBScript or other scripting languages

2.3 MSBuild

- MSBuild[3] is the command line based build system used by Visual Studio to compile .NET projects.
- Advantages of MSBuild
  - It can be used on a dedicated build machine without the Visual Studio.
  - It can be used to build several Visual Studio solutions together.
  - It can be used to perform additional tasks such as creating installers, archiving, retrieving from source control etc.
- **MSBuild** command :-(**Note**:- Underlined parameters are mandatory)
  - msbuild **{Path to Solution or Proj File}**
    /t:**{<u>Name of the targets to execute</u>}**
    /p:**{<u>Parameters to pass for building</u>}** /m:**{No. of concurrent processes}**
- **Possible Parameters**:-
  - Configuration :- Whether to build in build mode or release configuration
  - GENERATE_ASSEMBLYINFO :- Whether to generate assembly info after build or not
  - CodeContractsRunCodeAnalysis :- Boolean parameter
  - RunCodeAnalysis :-Boolean parameter
  - SUPRESS_CA :- Whether to suppress code analysis or not.
- **Example of MSBuild command**:
  - msbuild Target.proj /t:SampleTarget1;SampleTarget2 /m:1
    /p:Configuration=BuildConfiguration,GENERATE_ASSEMBLYINFO=0
    /p:CodeContractsRunCodeAnalysis=false,RunCodeAnalysis=Never,
    CodeContractsReferenceAssembly=DoNotBuild,SUPRESS_CA=1

2.4 Nagios Monitoring Tool

- **Introduction**:-
  - Nagios[5] is basically an open source monitoring tool,which is being used to monitor Servers,Applications and services.
  - It uses concept of continuous monitoring ,which can monitor network issues server issues to reduce maintenance cost.
- **Benefits of Nagios**:-
  - To get rid of periodic testing
  - Detects Split-Second failure
  - Reduces Maintenance Cost
  - Provides notification service incase of failure or breakdown
- **Prerequisites for Configuring Nagios**:-
  - Ubuntu 15.10
  - LAMP Stack on Ubuntu
  - Nagios Core
  - Nagios Plugins
  - SendEmail Plugin
  - NPRE Plugin
  - NSClient on Windows Server
- **Nagios Architecture**:

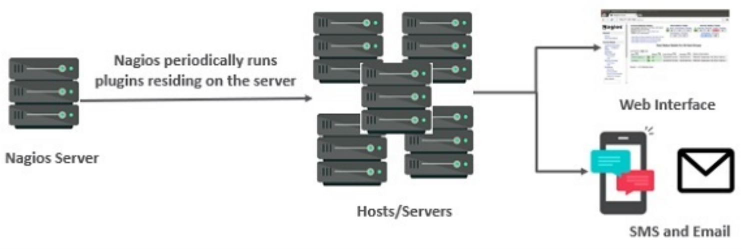


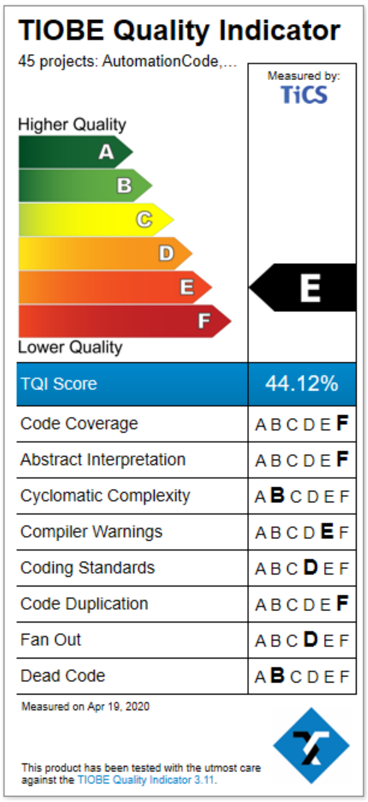
Figure 2.2: Nagios Architecture

2.5 Nessus Scanner

- Nessus[6] is basically a vulnerability scanner developed by tenable.io .Which is used to find the possible vulnerability in deployed system or in a target computer.
- Nessus employs Nessus Attack Scripting Language(NASL) to find the possible vulnerabilities, threats and potential attacks.
- **Nessus Capabilities**:-
  - Compatibility with computers and servers of all sizes.
  - Detection of security holes in remote computers and local host.
  - Detection of missing Security Mechanisms
  - Simulated attacks to pinpoint vulnerabilities.
  - Execution of security tests in a contained environment.
  - Scheduled security audits.
- Nessus Server is only available for Unix, Linux and FreeBSD. Where as Nessus Client is available for each and every OS

2.6 TICS Framework

- TICS[7] is a tool from Tiobe that measures **code quality** as TQI – **Tiobe Quality Indicator**.TQI metrics are mapped to the ISO 25010 software quality attributes.
- **TQI Matrices**
  - **Code Coverage**-Unit Test or Automated Test code coverage measured by specific coverage tools.
  - **Abstract Interpretation**- Deep flow analysis to detect all kinds of programming errors – Coverity.
  - **Cyclomatic Complexity**- Number of independent paths through a program.
  - **Compiler Warnings**- Measured by underlying complier with stricter options. TICS runs the compiler separately.


TIOBE Quality Indicator

45 projects: AutomationCode,...

Measured by: TiCS

TQI Score	44.12%
Code Coverage	A B C D E [✓ F]
Abstract Interpretation	A B C D E [✓ F]
Cyclomatic Complexity	A [✓ B] C D E F
Compiler Warnings	A B C D [✓ E] F
Coding Standards	A B C [✓ D] E F
Code Duplication	A B C D E [✓ F]
Fan Out	A B C [✓ D] E F
Dead Code	A [✓ B] C D E F

Measured on Apr 19, 2020

This product has been tested with the utmost care against the TIOBE Quality Indicator 3.11.

Figure 2.3: Nagios Architecture

- **Coding Standards**-Measured against the Philips coding standard.
- **Code Duplication**-TICS measures code duplication at file level and between files.
- **Fan Out**- Interdependency between modules.
- **Dead Code**- Measure of the unused Lines of Code.

• Significance of Different colors in TQI Dashboard

Category	**Name**	**TQI Score**
A	Outstanding	$\geq$90%
B	Good	$\geq$80%
C	Fairly Good	$\geq$70%
D	Moderate	$\geq$50%
E	Weak	$\geq$40%
F	Poor	$<$40%

Table 2.1: Significance of TQI Dashboard matrices

2.7 VMWare Work Station

- VMWare[8] work station is a tool which is developed by VMWare to run any operating system in a virtual environment created in hosted OS.
- Using this tool any OS with satisfying host System can be spinned up.
- To perform this operation one configuration file with extension .vmx is used to start/stop VM or to make any OS level configuration .
- .vmx file will next access .vmdk file which is responsible for handling the disk for particular VM.
- For the security reasons .lck file will be created when the VM will be turned on.

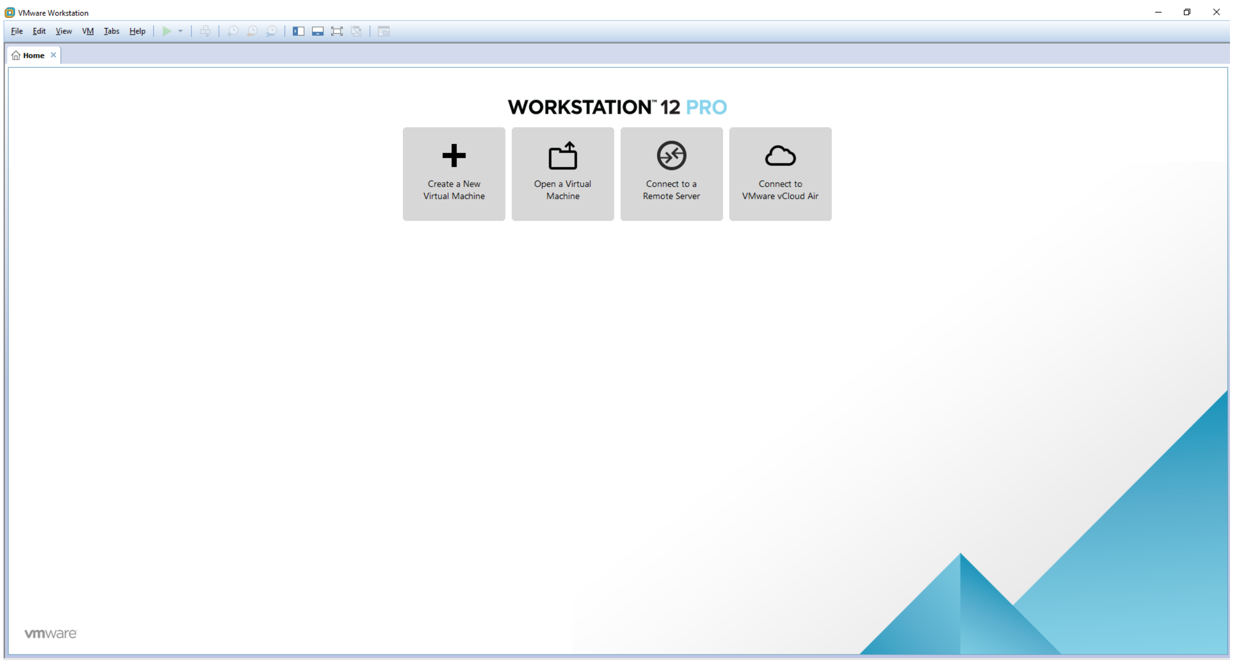

VMware Workstation
File Edit View VM Tabs Help
Home
WORKSTATION 12 PRO
Create a New Virtual Machine
Open a Virtual Machine
Connect to a Remote Server
Connect to VMware vCloud Air
vmware

Figure 2.4: VMWare Workstation

Chapter 3

Implementation

3.1 Mail Automation

- The task is to fetch the status of latest build and releases of particular application from TFS[1] server.
- For this task we can use azure URL to get all builds and releases details from TFS server
- URL for getting builds
  `https://dev.azure.com/organization/project/_apis/build/builds`
- URL for getting releases
  `https://dev.azure.com/organization/project/_apis/release/releases`
- after getting all builds and releases from TFS server,we need to filter it according to its id and map the latest builds with corresponding releases .
- After mapping form an HTML object and then after send it to the required persons.
- Steps involved in this task are as follow:
  - Get all build details from TFS server.Filter it according to project and branch.
  - Make HashMaps as below:
    * Map the build definition with corresponding application name.
    * Map the release definition with corresponding application name.

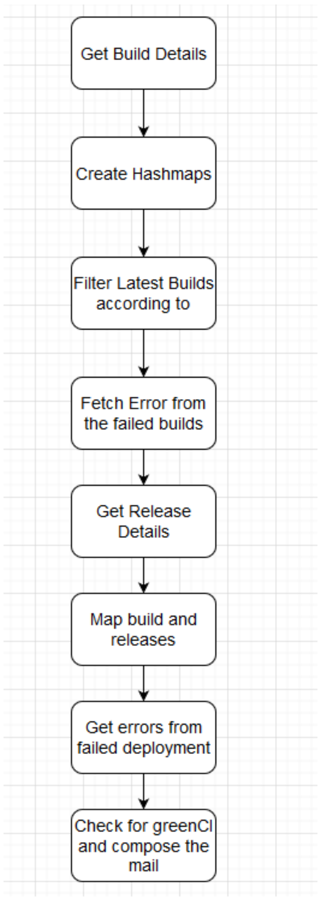
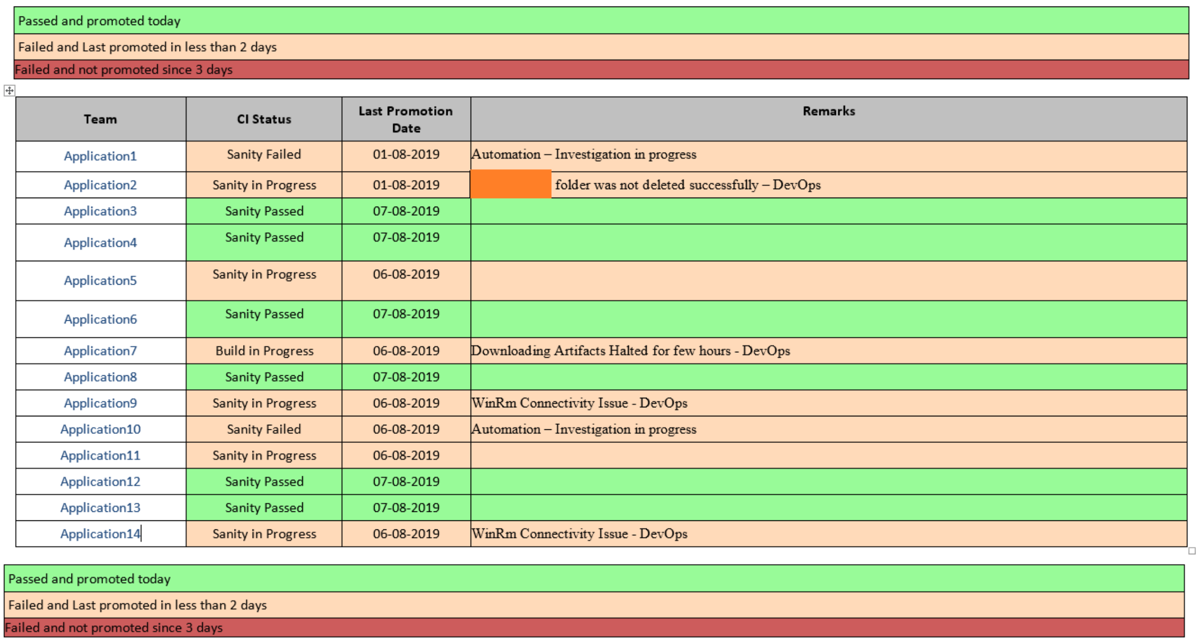

Get Build Details
Create Hashmaps
Filter Latest Builds according to
Fetch Error from the failed builds
Get Release Details
Map build and releases
Get errors from failed deployment
Check for greenCI and compose the mail

Figure 3.1: Flow of Mail Automation

  * Map Sanity log url with respective application name.

- Filter all builds according to their queue time and make sure that builds should be triggered less than a day.
- Fetch error for failed build from its error URL.
- Make release object from obtained releases
- Filter out the required releases.
- Map the builds and releases and store them to one object.
- Then Fetch error form each failed deployment of particular release.
- Check if all builds and releases are passed if passed mark it as green CI.
- Then form a mail object from collection.And send them to corresponding recipients.

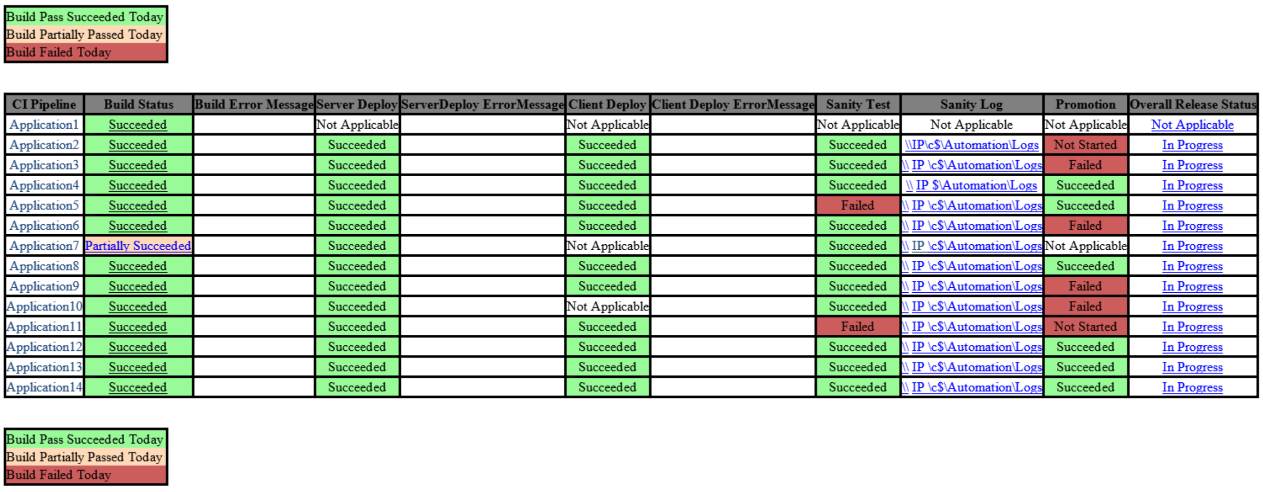


Passed and promoted today
Failed and Last promoted in less than 2 days
Failed and not promoted since 3 days

Team	CI Status	Last Promotion Date	Remarks
Application1	Sanity Failed	01-08-2019	Automation – Investigation in progress
Application2	Sanity in Progress	01-08-2019	folder was not deleted successfully – DevOps
Application3	Sanity Passed	07-08-2019	
Application4	Sanity Passed	07-08-2019	
Application5	Sanity in Progress	06-08-2019	
Application6	Sanity Passed	07-08-2019	
Application7	Build in Progress	06-08-2019	Downloading Artifacts Halted for few hours - DevOps
Application8	Sanity Passed	07-08-2019	
Application9	Sanity in Progress	06-08-2019	WinRm Connectivity Issue - DevOps
Application10	Sanity Failed	06-08-2019	Automation – Investigation in progress
Application11	Sanity in Progress	06-08-2019	
Application12	Sanity Passed	07-08-2019	
Application13	Sanity Passed	07-08-2019	
Application14	Sanity in Progress	06-08-2019	WinRm Connectivity Issue - DevOps

Passed and promoted today
Failed and Last promoted in less than 2 days
Failed and not promoted since 3 days

Figure 3.2: Manual Mail



Build Pass Succeeded Today
Build Partially Passed Today
Build Failed Today

CI Pipeline	Build Status	Build Error Message	Server Deploy	ServerDeploy ErrorMessage	Client Deploy	Client Deploy ErrorMessage	Sanity Test	Sanity Log	Promotion	Overall Release Status
Application1	Succeeded		Not Applicable		Not Applicable		Not Applicable	Not Applicable	Not Applicable	Not Applicable
Application2	Succeeded		Succeeded		Succeeded		Succeeded	\\IP\c$\Automation\Logs	Not Started	In Progress
Application3	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Failed	In Progress
Application4	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP $\Automation\Logs	Succeeded	In Progress
Application5	Succeeded		Succeeded		Succeeded		Failed	\\ IP \c$\Automation\Logs	Succeeded	In Progress
Application6	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Failed	In Progress
Application7	Partially Succeeded		Succeeded		Not Applicable		Succeeded	\\ IP \c$\Automation\Logs	Not Applicable	In Progress
Application8	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Succeeded	In Progress
Application9	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Failed	In Progress
Application10	Succeeded		Succeeded		Not Applicable		Succeeded	\\ IP \c$\Automation\Logs	Failed	In Progress
Application11	Succeeded		Succeeded		Succeeded		Failed	\\ IP \c$\Automation\Logs	Not Started	In Progress
Application12	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Succeeded	In Progress
Application13	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Succeeded	In Progress
Application14	Succeeded		Succeeded		Succeeded		Succeeded	\\ IP \c$\Automation\Logs	Succeeded	In Progress

Build Pass Succeeded Today
Build Partially Passed Today
Build Failed Today

Figure 3.3: Generated Mail

3.2 Developer Deploy

- The task is to make a customized build tool to help developers to test their developed applications on their own machine.
- The main file will be loading all the basic profiles and menus from json file and creating a GUI .

- Also from the main file all the relevant modules will be also loaded in parallel
- for creating the steps we need to have powershell[2] script and template to be loaded for each profile in a seperate folder.
- it is basically having features like Restore to defaults,Execute selected steps,Select all, Select None and Different steps like build,deploy,Package,Get Binaries from artifactory and more can be added as per requirement.
- Each steps will be calling its corresponding MSBuild[3] targets .

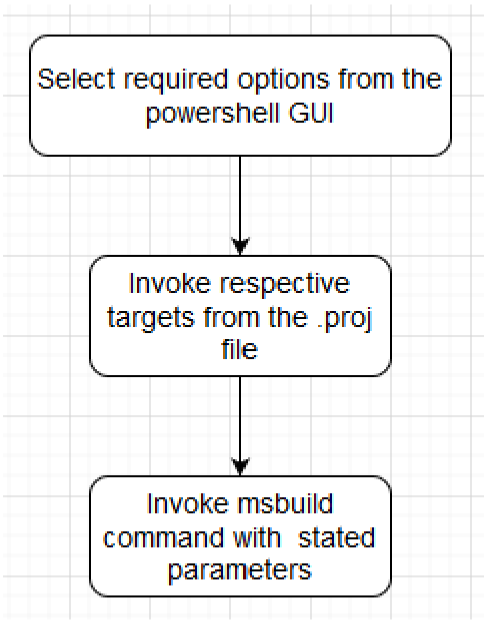
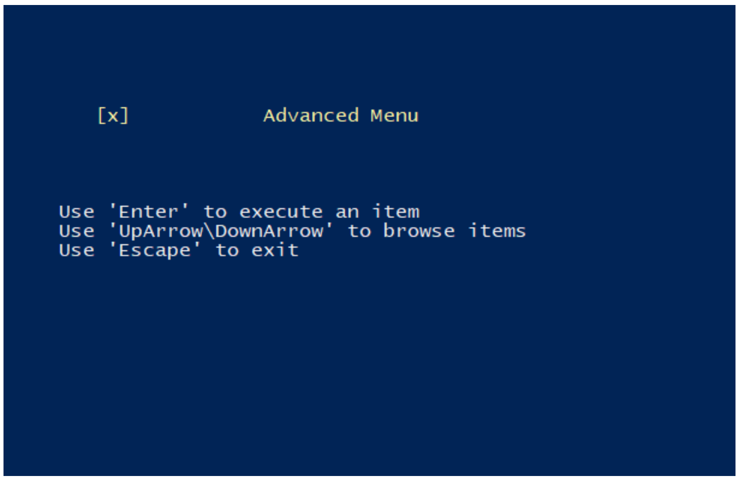

Select required options from the powershell GUI
Invoke respective targets from the .proj file
Invoke msbuild command with stated parameters

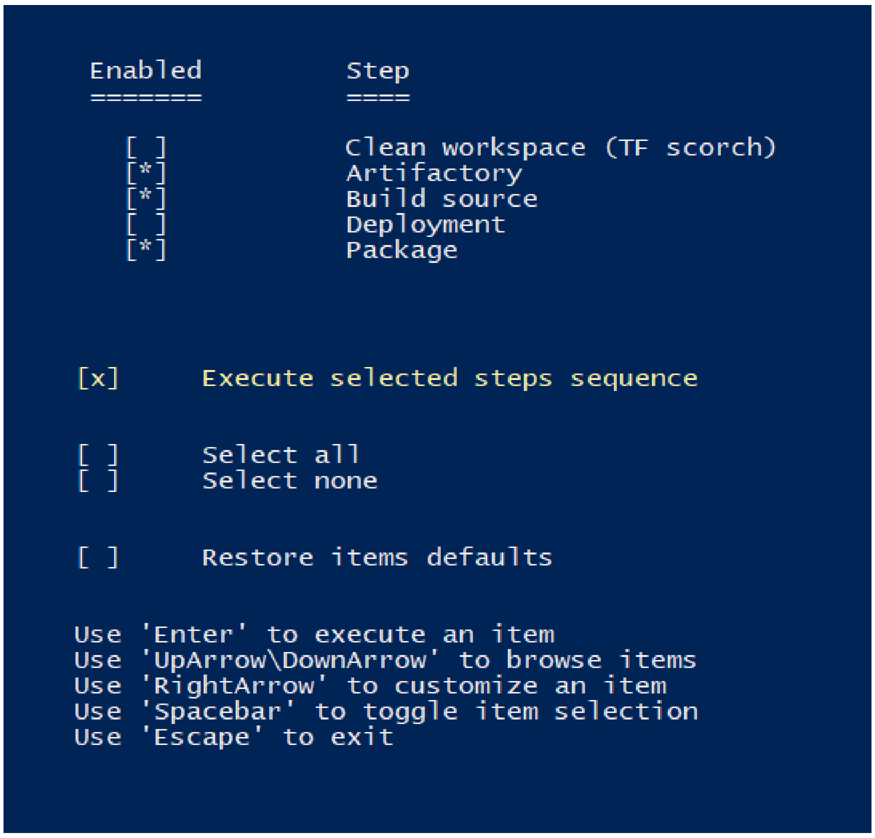
Figure 3.4: Design flow of Developer Deploy



```
   [x]           Advanced Menu


Use 'Enter' to execute an item
Use 'UpArrow\DownArrow' to browse items
Use 'Escape' to exit
```

Figure 3.5: Menu without Profiles



```
 Enabled          Step
 =======          ====

    [ ]           Clean workspace (TF scorch)
    [*]           Artifactory
    [*]           Build source
    [ ]           Deployment
    [*]           Package



[x]      Execute selected steps sequence


[ ]      Select all
[ ]      Select none


[ ]      Restore items defaults


Use 'Enter' to execute an item
Use 'UpArrow\DownArrow' to browse items
Use 'RightArrow' to customize an item
Use 'Spacebar' to toggle item selection
Use 'Escape' to exit
```

Figure 3.6: Steps of Advanced menu

3.3 Monitoring CI/CD Pipeline

- Tasks to be implemented on daily basis as shown in 2.1
  - Monitor Daily CI / CD pipeline[1].
  - Solve the issues coming in that pipeline
  - Verify and ensure the proper installation of product.
  - Release the end product for agile Testing.

3.4 Nagios Monitoring Tool

- Prerequisites are mentioned in 2.4 to configure nagios monitoring tool[5].
- Considering those prerequisites and following the below steps for installation of nagios on the ubuntu machine:
  - Install LAMP(Linux, Apache,MySql,PHP) Stack on Ubuntu.
  - Create a Nagios User and Group
  - Download and Install nagios and required plugins
  - Download and Install NRPE Plugins
- Once it is installed it can be accessed using **http://UbuntuIP/nagios** from another machines and **http://localhost/nagios** from the server itself
- The below configuration files are located in **/usr/local/nagios/etc**.
  - nagios.cfg
  - cgi.cfg
  - resource.cfg
  - commands.cfg
  - contacts.cfg
  - template.cfg
  - timeperiods.cfg

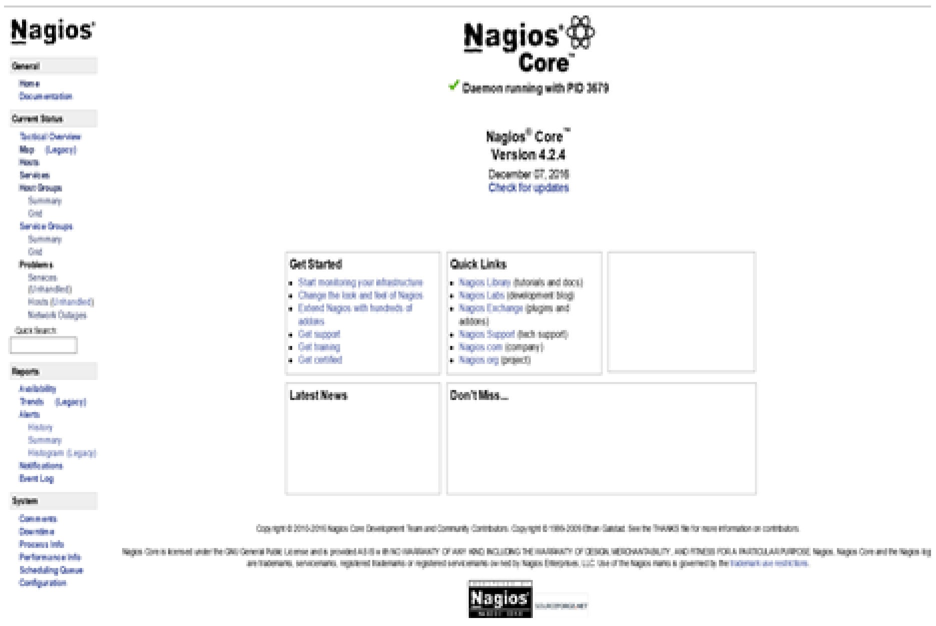
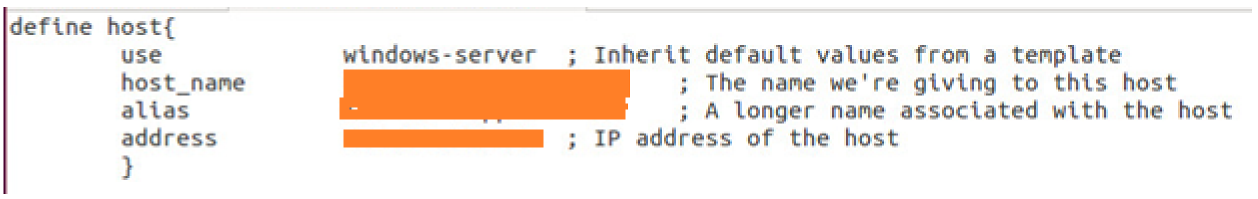


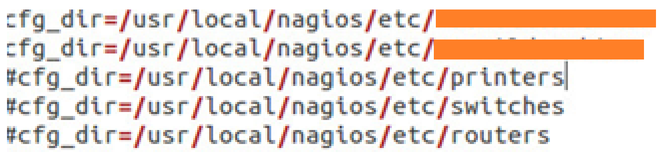
Figure 3.7: Nagios initial screen



```
define host{
        use             windows-server  ; Inherit default values from a template
        host_name       [redacted]          ; The name we're giving to this host
        alias           [redacted]          ; A longer name associated with the host
        address         [redacted]      ; IP address of the host
        }
```

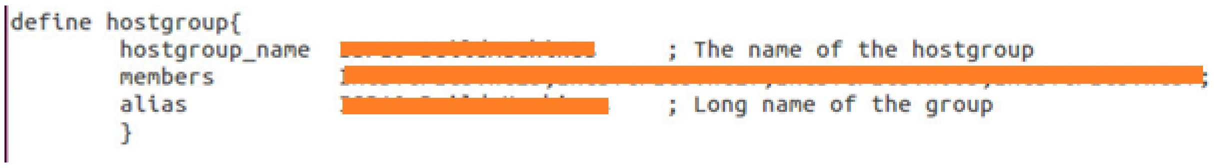
Figure 3.8: Template of configuring windows machine

- Monitoring on Windows machines:-
  - Adding hosts:- In Ubuntu hosts are defined in windows.cfg
    - **define host** - used to define a host in the configuration file.
    - **use** - is the namespace to allot the name of the template which is used by the host.
    - **host_name** - is the name of the host defined by the user.
    - **alias** - is the secondary or the common name of the host.
    - **address** - is the IP Address of the host.
- The hosts can also be added as separate files with the folder enabled in /usr/local/nagios/etc/nagios.cfg
- We can also define a set of hosts in specific groups known as hostgroup. A hostgroup is defined as follows:



```
cfg_dir=/usr/local/nagios/etc/[redacted]
cfg_dir=/usr/local/nagios/etc/[redacted]
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

Figure 3.9: Template of configuring hosts folders

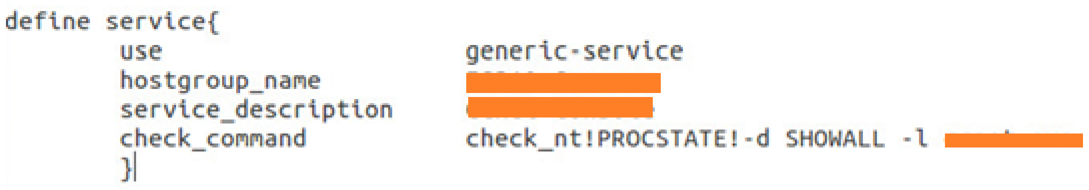


```
define hostgroup{
        hostgroup_name      [redacted]          ; The name of the hostgroup
        members             [redacted];
        alias               [redacted]          ; Long name of the group
        }
```

Figure 3.10: Template of configuring hosts

- **define hostgroup** - used to define a host group in the configuration file of the Nagios Server.
- **hostgroup_name** -name of the group of the hosts.
- **memebers** -used to define the members of the hostgroup, we specify the hostname of the machines in members.
- **alias** -is the common name which is used to define the group.

• Services that have to be monitored for the added hosts are defined as below:

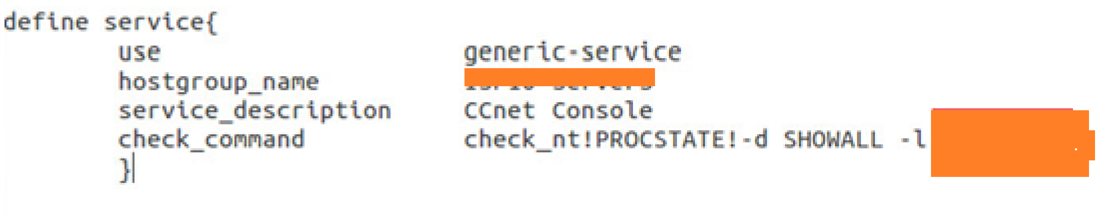


```
define service{
        use                     generic-service
        hostgroup_name          [redacted]
        service_description     [redacted]
        check_command           check_nt!PROCSTATE!-d SHOWALL -l [redacted]
        }
```

Figure 3.11: Adding Service monitoring

- **define service** is the syntax for defining service.
- **use** - used to define the templates for defining the service.
- **hostgroup_name** or **host_name** - name of the hostgroup for which this service is enabled.
- **service_description** is the name of the service which is show in the interface.
- **check_command** is the command which the service runs in order to monitor the service.

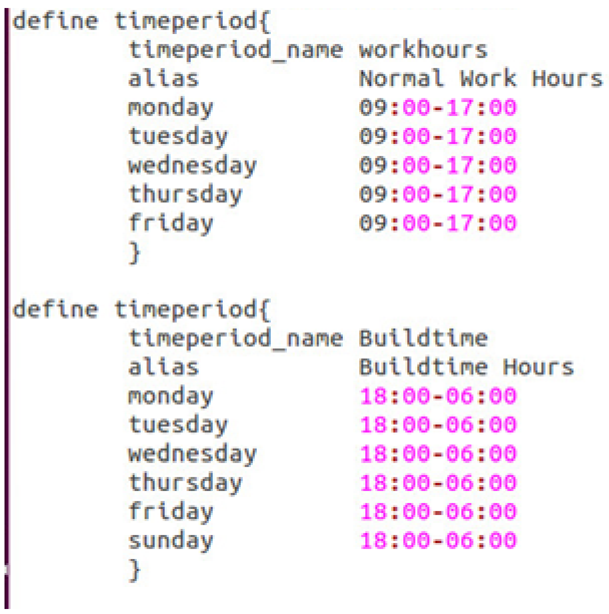
- Configuring Event handlers



```
define service{
        use                     generic-service
        hostgroup_name          [redacted]
        service_description     CCnet Console
        check_command           check_nt!PROCSTATE!-d SHOWALL -l [redacted]
        }
```

Figure 3.12: Event Handling

  - NRPE Plugin is used to start or stop any service or process on the windows servers
  - To enable the event handler for any service we add a **event_handler** attribute with name of the command which will run as an event handler as shown.
  - check_period is the time interval during which the service and the event handler will work. It is defined in timeperiod in **timeperiod.cfg** file as shown.

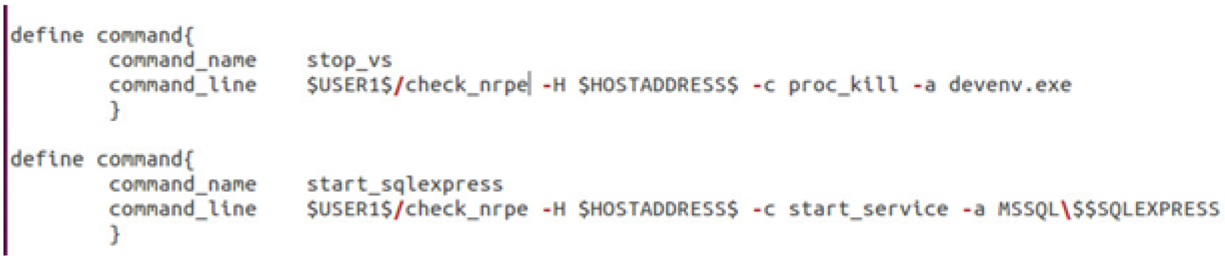


```
define timeperiod{
        timeperiod_name workhours
        alias           Normal Work Hours
        monday          09:00-17:00
        tuesday         09:00-17:00
        wednesday       09:00-17:00
        thursday        09:00-17:00
        friday          09:00-17:00
        }

define timeperiod{
        timeperiod_name Buildtime
        alias           Buildtime Hours
        monday          18:00-06:00
        tuesday         18:00-06:00
        wednesday       18:00-06:00
        thursday        18:00-06:00
        friday          18:00-06:00
        sunday          18:00-06:00
        }
```

Figure 3.13: Configuring Time period

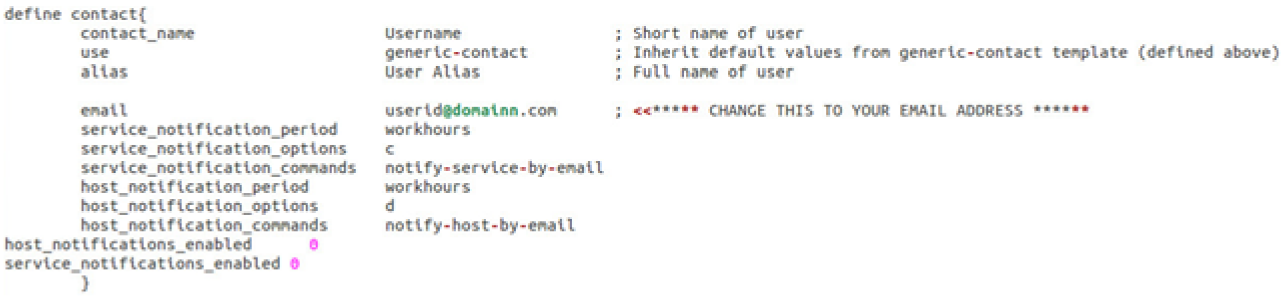
- The event handler command mentioned in the service definition is defined in **commands.cfg** file.
- **command_line** is the actual command which is executed when the event handler is triggered.



```
define command{
        command_name    stop_vs
        command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c proc_kill -a devenv.exe
        }

define command{
        command_name    start_sqlexpress
        command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c start_service -a MSSQL\$$SQLEXPRESS
        }
```

Figure 3.14: Defining Command



```
define contact{
        contact_name                    Username                ; Short name of user
        use                             generic-contact         ; Inherit default values from generic-contact template (defined above)
        alias                           User Alias              ; Full name of user

        email                           userid@domainn.com      ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
        service_notification_period     workhours
        service_notification_options    c
        service_notification_commands   notify-service-by-email
        host_notification_period        workhours
        host_notification_options       d
        host_notification_commands      notify-host-by-email
host_notifications_enabled      0
service_notifications_enabled 0
        }
```

Figure 3.15: Configuring Notifications

- **Configuring notifications and status mail**:-
  - Nagios has the in-built feature of notifying the administrator when any critical problem or recovery takes place.
  - in order to notify the administrator , one has to update **contacts.cfg** file in Nagios directory.
  - **define_contact** is the syntax which is used to define a contact in the notification. We give the contact name , email , service notification period etc. to define the contact.
- **Configuring Nessus client**:-
  - Once the Hosts are added. NSClient has to be installed on the windows Machine.(https://www.nsclient.org/download/)
  - With this set up Nagios will start monitoring that particular host.
  - The added hosts and host groups can be viewed in the hosts and Host Group section in the Nagios Web page.

- Once changes are made to the Nagios config files, **nagios.cfg** file has to compiled using the below command and Nagios service has to be restarted.
- Same can be done using following command:
  sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

3.5 STIG Implementation on Product

- **STIG** :- A Security Technical Implementation Guide is a cybersecurity methodology for standardizing security protocols within networks, servers, computers, and logical designs to enhance overall security. These guides, when implemented, enhance security for software, hardware, physical and logical architectures to further reduce vulnerabilities[4].
- To implement a powershell script got integrated containing all the changes with product installation containing this changes.
- **Features**:
  - **V-1089** :- Required legal notice must be configured to display before console logon.
    - Script required:
      - Set-ItemProperty -Path HKLM:\ SOFTWARE\ Microsoft\ Windows \CurrentVersion\ Policies\ System -Name LegalNoticeText -Value Message
      - Set-ItemProperty -Path HKLM:\SOFTWARE\ Microsoft\ Windows\ CurrentVersion\ Policies\ System -Name LegalNoticeCaption -Value Caption
      - Set-ItemProperty -Path HKLM:\ SOFTWARE\ Microsoft\ Windows\ CurrentVersion\ Policies\ System -Name PromptOnSecureDesktop -Value 1
  - **V-3383** :- The system must be configured to use FIPS-compliant algorithms for encryption, hashing, and signing.

    - ∗ Script required:
        - · Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy -Name Enabled -Value 1 -Type DWord
- – **V-1097** :- The number of allowed bad logon attempts must meet minimum requirements.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /lockoutthreshold:3”
- – **V-1098** :- The reset period for the account lockout counter must be configured to 15 minutes or greater on Windows 2012.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /lockoutwindow:15”
- – **V-1099** :- Windows 2012 account lockout duration must be configured to 15 minutes or greater.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /lockoutduration:30”
- – **V-1104** :- The maximum password age must meet requirements.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /maxpwage:60”
- – **V-1105** :- The minimum password age must meet requirements.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /minpwage:1”
- – **V-1107** :- The Password history must be configured to 24 passwords remembered.
    - ∗ Script required
        - · cmd.exe /c ”net accounts /uniquepw:24”
- – **V-6836** :- Passwords must, at a minimum, be 14 characters.
    - ∗ Script required:
        - · cmd.exe /c ”net accounts /minpwlen:14”

- **V-3449** :-Remote Desktop Services must limit users to one remote session.
  - Script required:
    - Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies \Microsoft\Windows NT\Terminal Services" -Name fSingleSessionPerUser -Value 1 -Type DWord
- **V-14249** :- Local drives must be prevented from sharing with Remote Desktop Session Hosts. (Remote Desktop Services Role).
  - Script required:
    - Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies \Microsoft\Windows NT\Terminal Services" -Name fDisableCdm -Value 1 -Type DWord
- **V-80477**:-Windows PowerShell 2.0 must not be installed on Windows 2012/2012 R2.
  - Script required:
    - Uninstall-WindowsFeature -Name PowerShell-v2
- **V-78059** :- Windows Server 2012/2012 R2 must be configured to audit Logon/Logoff - Account Lockout failures.
  - Script required:
    - cmd /c "AuditPol /set /category:"Logon/Logoff" /subcategory:"Account Lockout" /failure:enable /success:disable"
- **V-36707** :- Windows SmartScreen must be enabled on Windows 2012/2012 R2.
  - Script required:
    - Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies \Microsoft\Windows\System" -Name EnableSmartScreen -Value 1 -Type DWord
- **V-26602** :- The Microsoft FTP service must not be installed unless required.
  - Script required:
    - Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet \Services\ftpsvc" -Name Start -Value 3 -Type DWord

- **V-26576** :- The IP-HTTPS IPv6 transition technology must be disabled.
  - Script required:
    - Set-ItemProperty -Path ”HKLM:\Software\Policies\Microsoft \Windows\TCPIP\v6Transition\IPHTTPS\IPHTTPSInterface” -Name IPHTTPS_ClientState -Value 3 -Type DWord

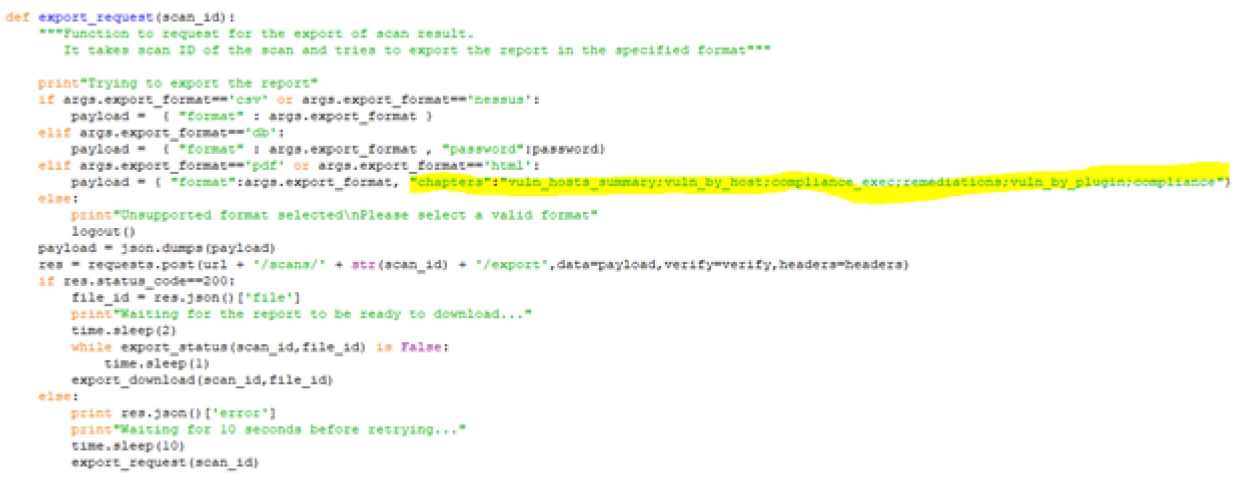
- **V-14234** :- User Account Control approval mode for the built-in Administrator must be enabled.
  - Script required:
    - Set-ItemProperty -Path ”HKLM:\Software\Microsoft\Windows \CurrentVersion\Policies\System” -Name FilterAdministratorToken - Value 1 -Type DWord
- **V-14235** :- User Account Control must, at minimum, prompt administrators for consent.
  - Script required:
    - Set-ItemProperty -Path ”HKLM:\Software\Microsoft\Windows \CurrentVersion\Policies\System” -Name ConsentPromptBehaviorAdmin -Value 4 -Type DWord
- **V-14236**:- User Account Control must automatically deny standard user requests for elevation.
  - Scripts required:
    - Set-ItemProperty -Path ”HKLM:\Software\Microsoft\Windows \CurrentVersion\Policies\System” -Name ConsentPromptBehaviorUser -Value 0 -Type DWord
- **V-14240** :- User Account Control must run all administrators in Admin Approval Mode, enabling UAC.
  - Script required
    - Set-ItemProperty -Path ”HKLM:\Software\Microsoft\Windows \CurrentVersion\Policies\System” -Name EnableLUA -Value 1 -Type DWord

3.6 Automated Nessus Scan

- **Nessus Installation**:-
  - Nessus[6] can be downloaded and installed from `https://www.tenable.com/downloads/nessus`
  - Make sure to install Nessus version **6.x**, because nessus versions on and after **7.x** are having **API Scans** disabled.
  - After installation the nessus can be activated in following two ways:
    1. **Online**:-
       * In the starting Screen after launching Nessus Web Client i.e. `http://localhost:8843`
         · Set up the proxy
         · Auth Method : AUTO DETECT
         · Then set the username and password to access the nessus client securely
         · If the validation of the activation key fails ,Then proceed with Offline Activation method.
    2. **Offline**:-
       * Go to `https://plugins.nessus.org/v2/offline.php` site
       * download plugins from the submission and download license file.
       * Place license file folder named conf inside Nessus folder under "C:\ProgramData\Tenable\Nessus"
       * Run cmd as admin
       * cd C:\ProgramData\Tenable\Nessus
       * then run command nessuscli fetch –register-offline "C:\ProgramData\Tenable \Nessus\conf\nessus.license"
       * After that update nessus with plugins using command nessuscli update all-2.0.tar
       * Then restart the system. After that you will be asked for credentials.

  - After successful activation of nessus disable the automatic update of all components.Just enable the automatic plugin update.

- **Nessus scan via API**:-
  - The nessus scan can be invoked via command line using API call.
  - One feasible API Script got found on github
    `https://github.com/AdmiralGaust/python-nessus`
  - After setting up all the prerequisite given on readme.md , we can launch API calls for nessus
    * python (path of the .py file) –list-policies (It will list all the policies)
    * python (path of the .py file) -t (target ip) -p ”policy_name” –e pdf –o “(path of the output)” –n “(Name of the scan , By default it will take Scan created by API Script)”
    * This scan will give the Host Executive summary which is much brief report.
    * If detailed report is needed the Modify the API Script as highlighted in the following image:



```
def export_request(scan_id):
    """Function to request for the export of scan result.
       It takes scan ID of the scan and tries to export the report in the specified format"""

    print"Trying to export the report"
    if args.export_format=='csv' or args.export_format=='nessus':
        payload =  { "format" : args.export_format }
    elif args.export_format=='db':
        payload =  { "format" : args.export_format , "password":password}
    elif args.export_format=='pdf' or args.export_format=='html':
        payload = { "format":args.export_format, "chapters":"vuln_hosts_summary;vuln_by_host;compliance_exec;remediations;vuln_by_plugin;compliance"}
    else:
        print"Unsupported format selected\nPlease select a valid format"
        logout()
    payload = json.dumps(payload)
    res = requests.post(url + '/scans/' + str(scan_id) + '/export',data=payload,verify=verify,headers=headers)
    if res.status_code==200:
        file_id = res.json()['file']
        print"Waiting for the report to be ready to download..."
        time.sleep(2)
        while export_status(scan_id,file_id) is False:
            time.sleep(1)
        export_download(scan_id,file_id)
    else:
        print res.json()['error']
        print"Waiting for 10 seconds before retrying..."
        time.sleep(10)
        export_request(scan_id)
```

Figure 3.16: Script modification

3.7 TICS Pipeline creation

- TICS is a tool which gives us detailed code analysis as mentioned and explained in 2.6[7].
- To configure project to work with TICS , its static path is needed to be configured with particular TICS project.
- **Setting up pipeline for TICS**:
  - First need to configure the relevant TFS agent to run on particular machine
  - Make sure the TICS project static path should be map to required location of the work folder.
  - Then schedule the build as per the requirement.
  - Build tasks are as follows:
    - Building the required solution using MSBuild task as follows:
      - Path of proj file
      - /t:Name of the target to build /m:4 /p:Configuration=(BuildConfiguration) ,GENERATE_ASSEMBLYINFO=0 /p:CodeContractsRunCodeAnalysis=false ,RunCodeAnalysis=Never,CodeContractsReferenceAssembly=DoNotBuild ,SUPRESS_CA=1
    - TICSQServer task for replicating and performing detailed code analysis.arguments to pass with this task as follows:
      - Project name
      - Branch name

3.8 Creation of automated build pipeline

- As mentioned in 3.7 the same procedure has been applied except TICS task for build configuration[1][3].
- Apart from this configuration, initial sanity tests can also be included as per developer's suggestion.

3.9 OVA Automation Implementation

- **Description**:-
  - Generally product is getting delivered in format of **OVA**.
  - Requirement from the management is to use VMWare Workstation[8] for creation of the OVA.
  - With that OVA file a template virtual machine containing installed Product with required demodata will be delivered.
- **Purpose**:-
  - Manually process will take almost 4 to 5 hrs of manual effort in creating OVA.
  - To reduce the manual effort automatic creation of OVA is proposed.
- **Prerequisite**:-
  - A command line tool **ovftool** should be installed.
  - Powershell[2] API named **vmxtoolkit** should be installed, if not then it can be installed using following command
    Install-Module -Name vmxtoolkit
- **Possible Approaches**:
  - There are three possible approaches to achieve this goal, which are mentioned as below:
    1. Create a one click OS ISO which is having all the automation scripts.
    2. After spinning OS or basic OVA template all the scripts will be copied to created new virtual machine and will trigger the installation automatically.
    3. To create an OVA template which is containing all the script and after spinning up the OS the installation will be automatically triggered.
- **Design according to selected approach**:-
  - Design for this automation can be divided into two parts, described as below:
    * **Level 1** :- This design is for host machine on which this script is running

    * **Level 2** :- This design is for the operations inside the virtual machine

- **Methodology**:
  - Creating OVA template :(Manual Activity mentioned in 3.17)
    * In VMWare workstation create a VM with required OS version and required specification.
    * Copy required scripts and folder in that VM on required locations
    * Enable **AutoLogon** for that VM and Increase **AutoLogonCount**
    * Map required shared locations.
    * PowerOff the VM,Unmount Operating system from the VM.
    * Edit .vmx file of that vm amd modify the vmconfiguration to 10 in place of 12.
    * Export that VM as OVA file.

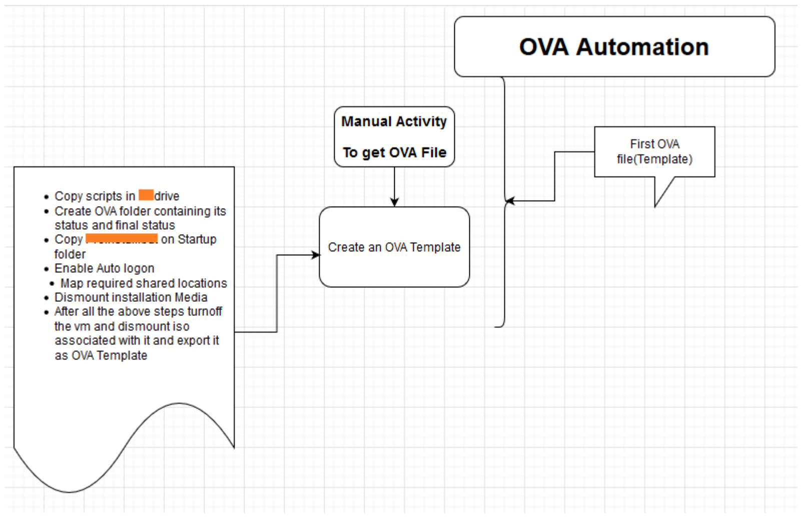

OVA Automation
Manual Activity
To get OVA File
First OVA file(Template)
Create an OVA Template
Copy scripts in [redacted] drive
Create OVA folder containing its status and final status
Copy [redacted] on Startup folder
Enable Auto logon
Map required shared locations
Dismount installation Media
After all the above steps turnoff the vm and dismount iso associated with it and export it as OVA Template

Figure 3.17: Creation of Template OVA

  - Automation: (Level 1 Mentioned in 3.18)
    * Using **ovftool template OVA** file need to convert to VMWare compatible file .
    * Then using vmxtoolkit API the VM can be invoked.
    * If the final status is success after installation then that VM will get exported to OVA which is the resultant OVA.

    * If the final status id failure after triggering installation than that VM will be powered off.
    * Logs will get generated for all relevant steps.
- Automation :(Level 2 Mentioned in 3.18)
    * After invocation of VM the file placed in startup location will be invoked and will take care of the product installation of the required version.
    * The product installation will be in silent mode.
    * After triggering product a script checking for successful product installation will be invoked.
    * once the product installation is successful userflags will be modified to enable autologon.
    * After successful product installation demodata installation will be triggred.
    * After successful installation all the relevant scripts will be removed and logs of Level 2 Automation will be removed after storing into shared location.
    * At the last the pagination will be setted up,Success status will be flagged and the VM will be restarted.

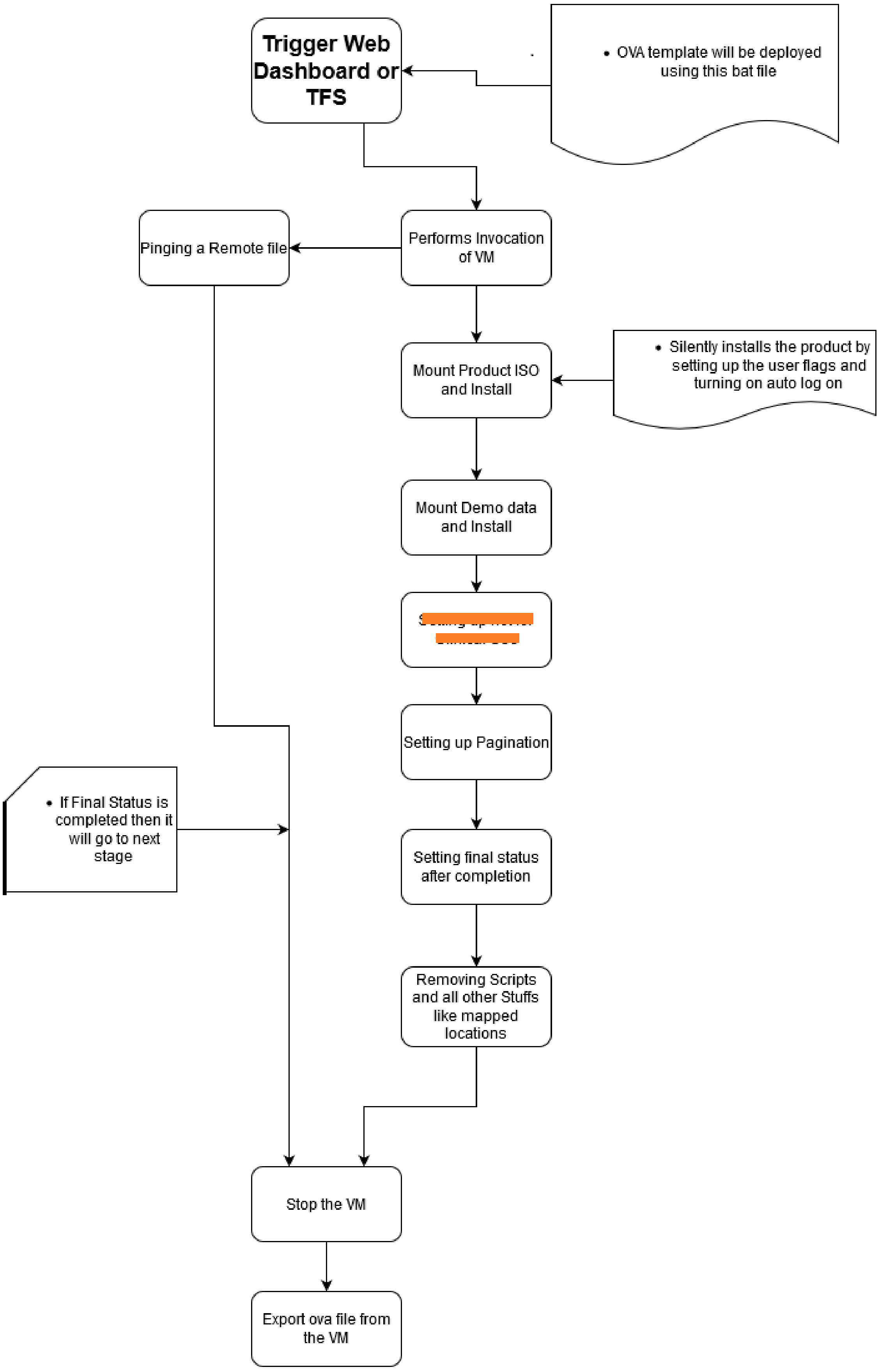

Trigger Web Dashboard or TFS
OVA template will be deployed using this bat file
Pinging a Remote file
Performs Invocation of VM
Mount Product ISO and Install
Silently installs the product by setting up the user flags and turning on auto log on
Mount Demo data and Install
[redacted]
Setting up Pagination
If Final Status is completed then it will go to next stage
Setting final status after completion
Removing Scripts and all other Stuffs like mapped locations
Stop the VM
Export ova file from the VM

Figure 3.18: Design of OVA creation

Chapter 4

Conclusion

4.1 Summary

From the above tasks it can be summarized that using powershell scripting,MSBuild and TFS as a basic tool to use for DevOps task ,it will be easier to implement automation and security in required application.

4.2 Conclusion

It can be concluded form the following project is that DevOps is having lot of scope of automation to deliver the product faster.Adding on to that this culture is useful and productive in case of faster delivery and bridging the gap between Developer and Operations team.Mail automation task will reduce the manual effort of surfing TFS dash board and will five the all possible errors in form of report.Developer deploy task will allow the developers to test their code in their own environment before proceeding for check-in .Maintaining CI/CD pipeline will make continuous delivery faster. With the help of nagios monitoring tool the required team can get a notification whenever the particular system or service will be down.STIG implementation will improve the security of the product after deployment.Nessus scan can give a detailed report of vulnerabilities or loop holes present in the machine after deployment of the product.OVA automation is reducing the 4 to 5 hrs of manual effort and it can give final OVA in one click.TICS pipeline will help to measure the code matrices.Thus the implementation of the above task will help to reduce manual effort,analyzing security and improving the product security.

References

[1] Micrsoft Team Foundation Server,
`https://docs.microsoft.com/en-us/azure/devops/server/tfs-is-now-azure-devops-server?view=azure-devops`

[2] Powershell-Scripting Microsoft Docs,
`https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6`

[3] MSBuild Microsoft Docs,
`https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild?view=vs-2019`

[4] Security Technical Implementation Guide ,
`https://www.stigviewer.com/stigs`

[5] Nagios Monitoring tool,
`https://www.nagios.org/`

[6] Nessus Scanner,
`https://www.tenable.com/products/nessus/nessus-professional`

[7] TICS Framework,
`https://www.tiobe.com/tics/tics-framework/`

[8] VMWare Work Station,
`https://www.vmware.com`