11-Nov-2020

GROUP MEMBERS

MANISH KUMAR KUMAWAT 2019HT66608

YOGESH TANWAR 2019HT66607

ISHWAR 2019HT666

SANJAY DAHIYA 2019HT666

DevOps Assignment

Hands on experience of Git Workflow

Contents

[1 Executive Summary 2](#_Toc55732623)

[2 Create a Repository and installation of Git-Bash 3](#_Toc55732624)

[2.1 Creation of New Repository 3](#_Toc55732625)

[2.2 Installation of Git Bash 4](#_Toc55732626)

[3 Add team members as collaborators and assign them appropriate rights (if you are performing individual then create one dummy account for the task) 5](#_Toc55732627)

[3.1 Current configuration of Data Connectors in Sentinel 5](#_Toc55732628)

[3.2 Recommendations: 5](#_Toc55732629)

[4 Create a branch(development/production/feature) 5](#_Toc55732630)

[4.1 Current Active Rules 5](#_Toc55732631)

[4.2 Recommendation Custom Analytics Rules: 5](#_Toc55732632)

[5 Edit files or create new files followed by commit 6](#_Toc55732633)

[6 Clone the repo and Create pull-request 7](#_Toc55732634)

[6.1 Workbooks in current configuration 7](#_Toc55732635)

[6.2 Recommendation: 7](#_Toc55732636)

[7 While collaborating your work, showcase how conflicts are resolved 7](#_Toc55732637)

[7.1 Prerequisites 7](#_Toc55732638)

[7.2 Incident 1 8](#_Toc55732639)

[7.3 Incident2 8](#_Toc55732640)

[7.4 Closing an incident 8](#_Toc55732641)

[8 Create tag such as open issue, or feature-added 8](#_Toc55732642)

[9 Do a force push and then later reset the changes 8](#_Toc55732643)

[10 Stage “development branch to production branch” 9](#_Toc55732644)

[11 Showcase how features are released in versions (merging production to master branch) 9](#_Toc55732645)

[12 State importance of Readme and gitignore files and their usage while working in a distributed environment. 10](#_Toc55732646)

# Executive Summary

The objective of this assignment is to have hands on experience of Git Workflow. You are required to create or fork a repo on Git and use Git bash on the desktop for completing the tasks. You are required to create one manager role for the repository and others as collaborators. The manager will act as a reviewer and decision maker.

Checklist of the tasks that are required to be performed:

* Add team members as collaborators and assign them appropriate rights (if you are performing individual then create one dummy account for the task)
* Create a branch (development/production/feature)
* Edit files or create new files followed by commit
* Clone the repo and Create pull-request
* While collaborating your work, showcase how conflicts are resolved
* Create tag such as open issue, or feature-added
* Do a force push and then later reset the changes
* Stage “development branch to production branch”
* Showcase how features are released in versions (merging production to master branch)
* Also, state importance of Readme and gitignore files and their usage while working in a distributed environment.

Bonus point: If Pull Requests are linked with e-mail to the manager who finally approves the changes.

# Create a Repository and installation of Git-Bash

We can store a variety of projects in GitHub repositories, including open source projects. With open source projects, we can share code to make better, more reliable software.

## Creation of New Repository

In the upper-right corner of any page, use the drop-down menu, and select New repository.

Graphical user interface, application

Description automatically generated

After Selecting a new repository, we can put any name and set privacy for it.

Graphical user interface, text, application, email

Description automatically generated

## Installation of Git Bash

To install Git Bash for desktop go to <https://git-scm.com/download/win> and select windows setup accordingly.

Graphical user interface, text, application

Description automatically generated

To setup username and email use following commands

git config –global user.name “manishbits”

git config –global user.email 2019HT66608@wilp.bits-pilani.ac.in

Text

Description automatically generated

# Add team members as collaborators and assign them appropriate rights (if you are performing individual then create one dummy account for the task)

## Add team members as collaborators

## Assign Appropriate rights to contributors

As per standard practice we should be looking at adding few more important data connectors as below:

# Create a branch(development/production/feature)

Azure Sentinel uses [**analytics to correlate alerts into incidents**](https://docs.microsoft.com/en-us/azure/sentinel/tutorial-detect-threats-built-in). **Incidents** are groups of related alerts that together create an

## Current Active Rules

## Recommendation Custom Analytics Rules:

Here are few more important analytics rules as below:

# Edit files or create new files followed by commit

Azure Sentinel powerful hunting search and query tools to hunt for security threats across organization's data sources. To help security analysts look proactively for new anomalies that were not detected by security apps, Azure Sentinel' built-in hunting queries guide into asking the right questions to find issues in the data are already have on the network.

# Clone the repo and Create pull-request

Built-in workbooks provide integrated data from connected data sources to let deep dive into the events generated in those services. The built-in workbooks include Azure AD, Azure activity events, and on-premises, which can be data from Windows Events from servers, from first party alerts, from any third-party including firewall traffic logs, Office 365, and insecure protocols based on Windows events. The workbooks are based on Azure Monitor Workbooks to provide with enhanced customizability and flexibility in designing the custom workbook.

## Workbooks in current configuration

## Recommendation:

Here are few nice to have workbook configuration as below:

# While collaborating your work, showcase how conflicts are resolved

An incident can include multiple alerts. It is an aggregation of all the relevant evidence for a specific investigation. An incident is created based on analytics rules that created in the **Analytics** page.

## Prerequisites

* It requires to use the entity mapping fields in the setup of analytic rule before investigating the incident. The investigation graph requires the original incident includes entities.
* If there is a guest user that needs to assign incidents, the user must be assigned the Directory Reader role in Azure AD tenant. Regular (non-guest) users have this role assigned by default.

## Incident 1

* 1. The Incident page has information for total incidents with status as: open, In Progress & Closed. The status of each incident can be checked on this incident page along with the priority.
  2. User can filter the incidents as needed, for example by status or severity.
  3. Incidents can be assigned to a specific user. For each incident it can be assigned to owner, by setting the Incident owner field. All incidents start as unassigned. A user can also add comments so that other analysts will be able to understand what its investigated and what are concerns around the incident.

## Incident2

The investigation graph enables analysts to ask the right questions for each investigation. The investigation graph helps to understand the scope, and identify the root cause, of a potential security threat by correlating relevant data with any involved entity. It can dive deeper and investigate any entity presented in the graph by selecting it and choosing between different expansion options.

## Closing an incident

Once user has resolved a particular incident (for example, when investigation has reached its conclusion),

# Create tag such as open issue, or feature-added

## Closing an incident

# Do a force push and then later reset the changes

## Closing an incident

# Stage “development branch to production branch”

# Showcase how features are released in versions (merging production to master branch)

# State importance of Readme and gitignore files and their usage while working in a distributed environment.