Name: Peñas, Issa Victoria H.	**Date Performed:** 10/21/2022
Course/Section: CPE232 - CPE31S22	**Date Submitted:** 10/23/2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Semester (SY: 2021 - 2022)

Activity 9: Install, Configure, and Manage Performance Monitoring tools

1. Objectives

Create and design a workflow that installs, configure and manage enterprise performance tools using Ansible as an Infrastructure as Code (IaC) tool.

2. Discussion

Performance monitoring is a type of monitoring tool that identifies current resource consumption of the workload, in this page we will discuss multiple performance monitoring tool.

Prometheus

Prometheus fundamentally stores all data as timeseries: streams of timestamped values belonging to the same metric and the same set of labeled dimensions. Besides stored time series, Prometheus may generate temporary derived time series as the result of queries. Source: Prometheus - Monitoring system & time series database

Cacti

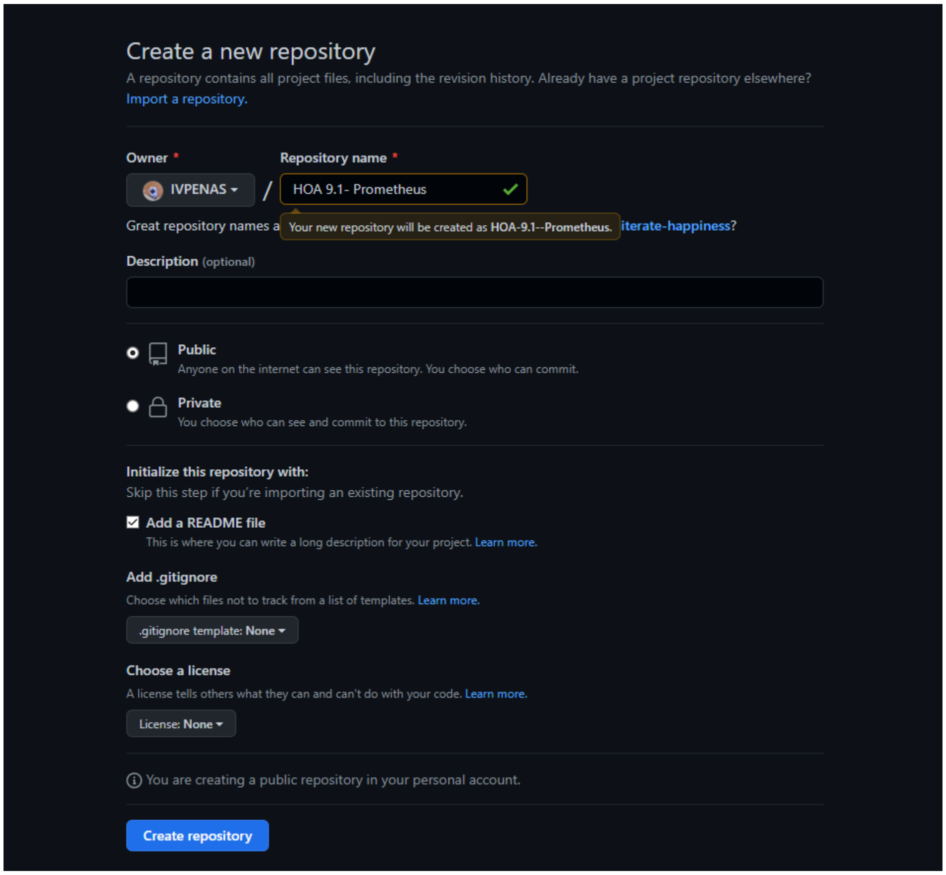
Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with thousands of devices. Source: Cacti® - The Complete RRDTool-based Graphing Solution

3. Tasks

1. Create a playbook that installs Prometheus in both Ubuntu and CentOS. Apply the concept of creating roles.
2. Describe how you did step 1. (Provide screenshots and explanations in your report. Make your report detailed such that it will look like a manual.)
3. Show an output of the installed Prometheus for both Ubuntu and CentOS.
4. Make sure to create a new repository in GitHub for this activity.

4. Output (screenshots and explanations)

1. Create a new repository in GitHub under the name of **HOA-9.1- Prometheus**, and make sure that the repository is **Public.** As a good practice add a **READ.md** file and input any related information regarding your inserted repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * IVPENAS / Repository name * HOA 9.1- Prometheus ✓

Great repository names a Your new repository will be created as HOA-9.1--Prometheus. iterate-happiness?

Description (optional)

● Public — Anyone on the internet can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

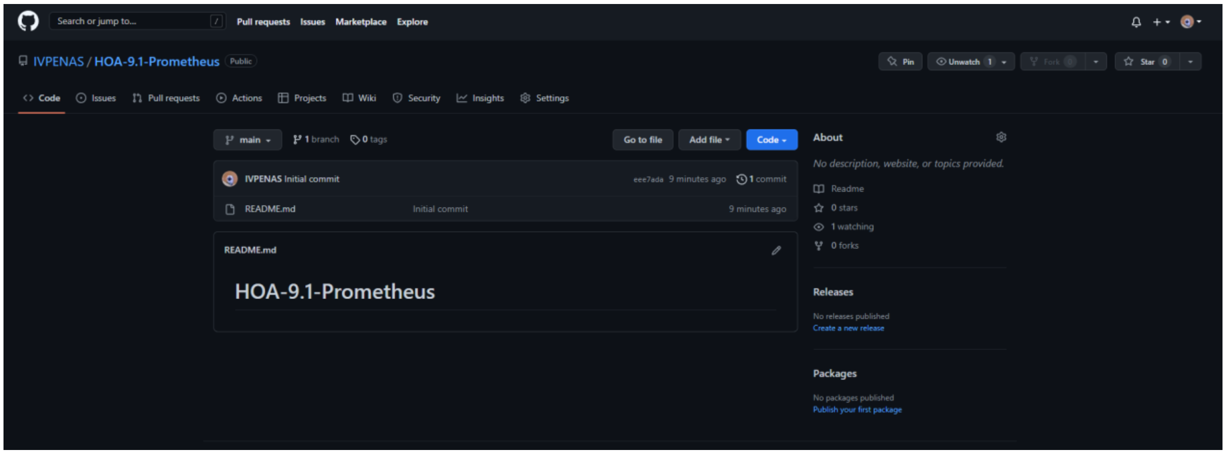
☑ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

ⓘ You are creating a public repository in your personal account.

Figure 1.1. Shows the creation of New Repository in Github



IVPENAS / HOA-9.1-Prometheus Public

main · 1 branch · 0 tags

IVPENAS Initial commit — eee7ada 9 minutes ago — 1 commit
README.md — Initial commit — 9 minutes ago

README.md

HOA-9.1-Prometheus

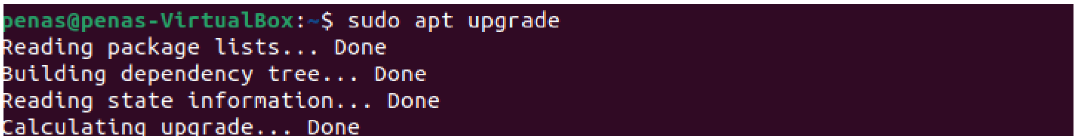
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Figure 1.2. Shows the Created Repository named HOA-9.1-Prometheus

2. Make sure that your Local Server was up to date, if not execute the commands: **sudo apt update**, and **sudo apt upgrade.**



```
penas@penas-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
```

Reflections:

Answer the following:

1. What are the benefits of having a performance monitoring tool?

   Defining a Performance Monitoring Tool is a type of application that **monitors, observes, identifies, collects, and reports** the whole infrastructure of a Network or a Computer System in which it mainly focuses on **cloud applications, log issues,** and **more,** relating to the overall operational health of a System. Most Operating systems have built-in Performance Monitor typically the data collected will be either displayed using **graphs** or a **series of percentages** from *CPU, memory, hard disk, and networks* which **notify and aid the System Administrators to monitor the current state** and **identify future events** specifically if a sector needs upgrading to ensure if the networks are efficient. It has also a feature to provide suggestions and guidelines in order to improve the system's performance by tuning or optimizing the systems. Prometheus was one of the Performance Monitoring Tools which detects the source of errors and notifies the System Administrators while monitoring the System, sadly it has no feature to fix errors or bugs but on the contrary, it'll obtain [1] performance-related information from the OS resulting in the Application to [2] store its data for future use in problem diagnosis, [3] Tracking changed that are made to the system and application and more, from continuously using the Performance Monitoring Tool.

Conclusions:

The students were introduced to the Performance Monitoring Tool which is a type of application that monitors, observes, identifies, collects, and reports the current activity and state of a Network or Computer System, whereas it allows the System Administrator to [1] monitor multiple states of a server including the data from their CPU, memory, hard disk, and networks representing in graphs or percentage, [2] it'll notify if there's an event that needed to be resolved, [3] due to its continuous use, it'll store the previous data and refer to them for future use specifically in problem diagnosis. Prometheus is one example of a Performance Monitoring Tool that was applied and installed using an ansible playbook where the student was able to accomplish and verify, it'll take a bit of time for the student to get used to the specific formats of the ansible-playbook as the pupil was able to solve some errors to optimize the efficiency of the using the playbook itself.