| **Name:** Torrecampo, Juan Piolo S. | **Date Performed:** Oct 22, 2022 |
| --- | --- |
| **Course/Section:** CPE 232 / CPE31S22 | **Date Submitted:** Oct 22, 2022 |
| **Instructor:** Dr. Jonathan Taylar | **Semester and SY:** 1st Sem, 2022 - 2023 |
| **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. | |
| 1. **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](https://prometheus.io/)  **Cacti**  Cacti is a complete network graphing solution designed to harness the power of [RRDTool](http://www.rrdtool.org/)'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](https://www.cacti.net/info/cacti) | |
| 1. **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. | |
| 1. **Output** (screenshots and explanations) | |
| **STEP 1. Create a new repository and clone it into your machine.**    *Figure 1.1. Github page of the new repository.*    *Figure 1.2. Cloning the newly created repository in the host machine.*  **STEP 2. Copy the inventory and ansible.cfg from the previous repository in Activity 8*.***    *Figure 1.3. Copying the inventory and ansible.cfg from the previous repository.*  **STEP 3. Create the needed files and directories, as shown in *Figure 1.4.***    *Figure 1.4. Tree structure of the directory.*          *Figure 1.5. Creating the files and directories inside of the repository*  **STEP 4. Copy the format below for the inventory file. Take note that the IP address and username varies depending on your machine.**    *Figure 1.6. Content of inventory file which supports the 2 servers (Ubuntu and CentOS).*  **STEP 5. Copy the contents of the install\_prometheus.yml below.**      *Figure 1.7. Editing the install\_nagios.yml.*  **STEP 6. Copy the contents of the main.yml for Centos and Ubuntu.**      *Figure 1.8. The screenshot above shows the contents of main.yml under centos.*  **STEP 7. In the file prometheus.service, copy the contents below and paste to the file. This file is a service file that enables the prometheus service to run after booting up.**    *Figure 1.9. The screenshot above shows the contents of prometheus.service under files directory.*  **Step 8. Check the connection between the computer and the servers by running “ansible all -m ping” (Note: ssh without asking a password is required). Also, you can check you syntax by running “ansible-playbook –syntax-check playbook.yml”**    *Figure 1.10. Running a test to make sure the two servers are reachable.*    *Figure 1.11. Running a test to make sure the scripts does not have any syntax errors.*  **Step 9. Run “ansible-playbook –ask-become-pass playbook.yml” to execute the scripts to the servers.**      *Figure 1.12. Output of the command “ansible-playbook –ask-become-pass install\_nagios.yml”.*  **Step 10. Verify the installation by checking both servers. In firefox or other web browser, type the IP address of the machine and the port 9090 (e.g, 192.168.122.94:9090). The port 9090 is where the Prometheus used in connecting the website.**   | **VERIFYING PROMETHEUS IN BOTH SERVER** | | | --- | --- | | **Server** | **Output in Firefox** | | **CentOS**  *(192.168.122.190:9090)* |  | | **Ubuntu**  *(192.168.122.94:9090)* |  |   *Table 1.1. The table above shows the proofs in each server.*  **Step 11. Update the repository.**    *Figure 1.13. Pushing the updated repository into Github.*    *Figure 1.14. The screenshot above shows the Github updated repository page.*  **Github Link:** <https://github.com/piolotorrecampo/CPE232-Activity_9.git>  **Prometheus Download Page:** <https://prometheus.io/download/> | |
| **Reflections:**  Answer the following:   * 1. What are the benefits of having a performance monitoring tool? * Having a performance monitoring tool is a life saver for the system administrations. It gives access to the system administrators to monitor the current state of the servers. This monitors whether the system has errors, enough hardware resources, processes, and many more. One example that supports this massive feature is the Prometheus. Prometheus can detect the source of errors and alerts the system administrators and constantly monitor services per device but is not responsible for fixing bugs or errors. The Cacti is not included in the activity but upon researching, the Cacti is also considered as a monitoring tools but has a feature of creating performance management Graphs. | |
| **Conclusions:**  This activity achieves its goal to introduce other enterprise performance tools that apply the use of Ansible as a medium in automating the processes in a server. In the procedure part of the activity introduces the monitoring tool called Prometheus. The first step in completing this activity is to research for a possible installation guide for a Ubuntu and CentOS server. Then convert this set of instructions into a ansible playbook that installs and runs services in both servers. Upon observation throughout this activity, I realized that the method of installing and starting the services of Prometheus in both servers are the same. Overall, this activity gives me confidence in creating ansible playbooks and to implement roles to it. | |
| **Honor Pledge:**  *“I affirm that I will not give or receive unauthorized help on this activity and that all will be my own.”* | |