Overview of Project VM Setup

- **Provisioning of VMs**

  Four virtual machines are to be set up for various services.

- **VM1: Jenkins Server**

  The first VM will host the Jenkins CI/CD server for automation.

- **VM2: Gogs Server**

  The second VM will run the Gogs server for Git management.

- **VM3: Web Server (Apache)**

  The third VM will serve as a web server using Apache.

- **VM4: Monitoring Server**

  The fourth VM will include Grafana for monitoring services.

User Management Process on VM3

- **Creating Users on VM3**

  A bash script creates three users: Devo, Testo, and Prodo on VM3.

- **Centralized Access Control**

  All three users are added to a group named 'deployG' for simplified access control.

- **User Deletion Script Implementation**

  A script will allow deletion of a user by providing the username as an argument.

Integrating Gogs with Jenkins for CI/CD

- **Automated Deployment Processes**

  Integrating Gogs with Jenkins automates deployment processes via webhooks.

- **Webhook Triggers**

  Set up webhooks in Gogs to trigger Jenkins jobs seamlessly.

- **Streamlined CI/CD Pipeline**

  Enhances the CI/CD pipeline by automating code deployment with Gogs and Jenkins.

- **Efficient Code Management**

  Gogs facilitates efficient code management while Jenkins handles the automation.

- **Continuous Integration Benefits**

  This integration ensures continuous integration and delivery, reducing manual errors.

- **Collaboration Enhancement**

  Improves team collaboration by automating deployment tasks.

- **Error Reduction**

  Automated processes minimize errors compared to manual deployments.

Creating a Git Repository on Gogs

- **Develop Ansible Playbook for Apache**

  Create InstallApache.yml to automate Apache installation on VM3, ensuring it runs properly.

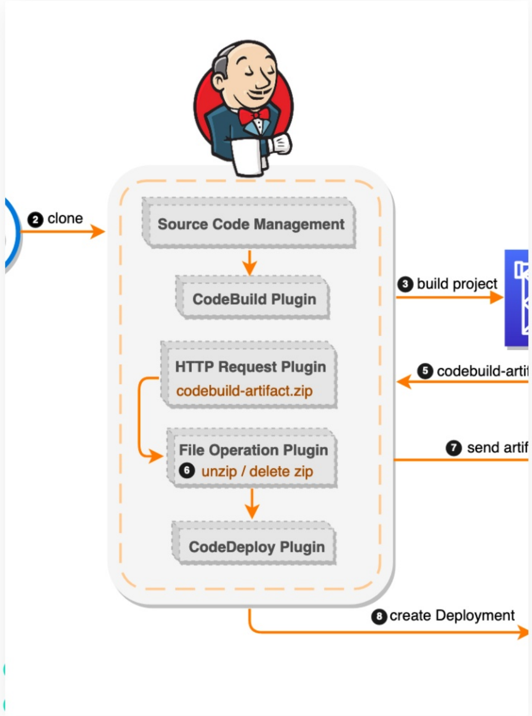
- **Script to Identify User Groups**

  Implement NotGroupMembers.sh to list users not part of the 'deployG' group, enhancing user management.

- **Grafana Setup with Ansible**

  Design SetupGrafana.yml to automate Grafana installation and configuration on VM4, streamlining monitoring.

Configuring CI/CD Pipeline with Jenkins



- **Integrate Jenkins with Gogs**

  Set up Jenkins to monitor the Gogs repository for changes to trigger builds automatically.

Automating Jenkins CI/CD Pipeline

Streamlining CI/CD Processes with Jenkins and Ansible

- **Automated Code Commit Detection**

  The Jenkins pipeline triggers on code commits for immediate action.

- **Ansible Playbook Execution (Apache)**

  Executes InstallApache.yml to install Apache on VM3.

- **Docker Image Build and Archive**

  Builds a Docker image from Dockerfile, saves it locally, and archives it.

- **Email Notification for Pipeline Status**

  Sends email with pipeline execution status and user details in 'deployG' group.

- **Timestamp in Notifications**

  Includes date and time of the pipeline execution in email notifications.

- **Docker Image Path Included**

  Email features the path to the generated Docker image tar file.

- **Ansible Playbook Execution (Grafana)**

  Triggers SetupGrafana.yml to install Grafana on VM4 after Docker operations.

- **Separate Grafana Setup Notification**

  Sends a distinct email with Grafana setup status and dashboard URL.

Essential Configuration Guidelines for CI/CD

Key steps for setting up your CI/CD pipeline

- **Firewall Configuration**

  Ensure to configure firewall rules or port forwarding as needed for secure access.

- **Operating System Requirements**

  Make sure the servers are running either CentOS or RockyLinux for compatibility.

- **Email Notification Options**

  Choose your preferred method for email notification: Bash, Ansible, or Jenkins.

Key Steps for Project Completion

- **Git Repository Setup**

  Share a git repo with Three3mr as collaborator, including essential files.

- **Include Bash Scripts**

  Ensure bash scripts are part of the repository to automate tasks.

- **Ansible Playbooks**

  Add Ansible playbooks that define the automation configuration.

- **Jenkinsfile Stages**

  Include a Jenkinsfile detailing the build and deployment stages.

- **Documentation with README**

  Provide a README documenting the Ansible playbooks and Jenkins pipeline.

- **Presentation Submission**

  Share a presentation showcasing your understanding and implementation.