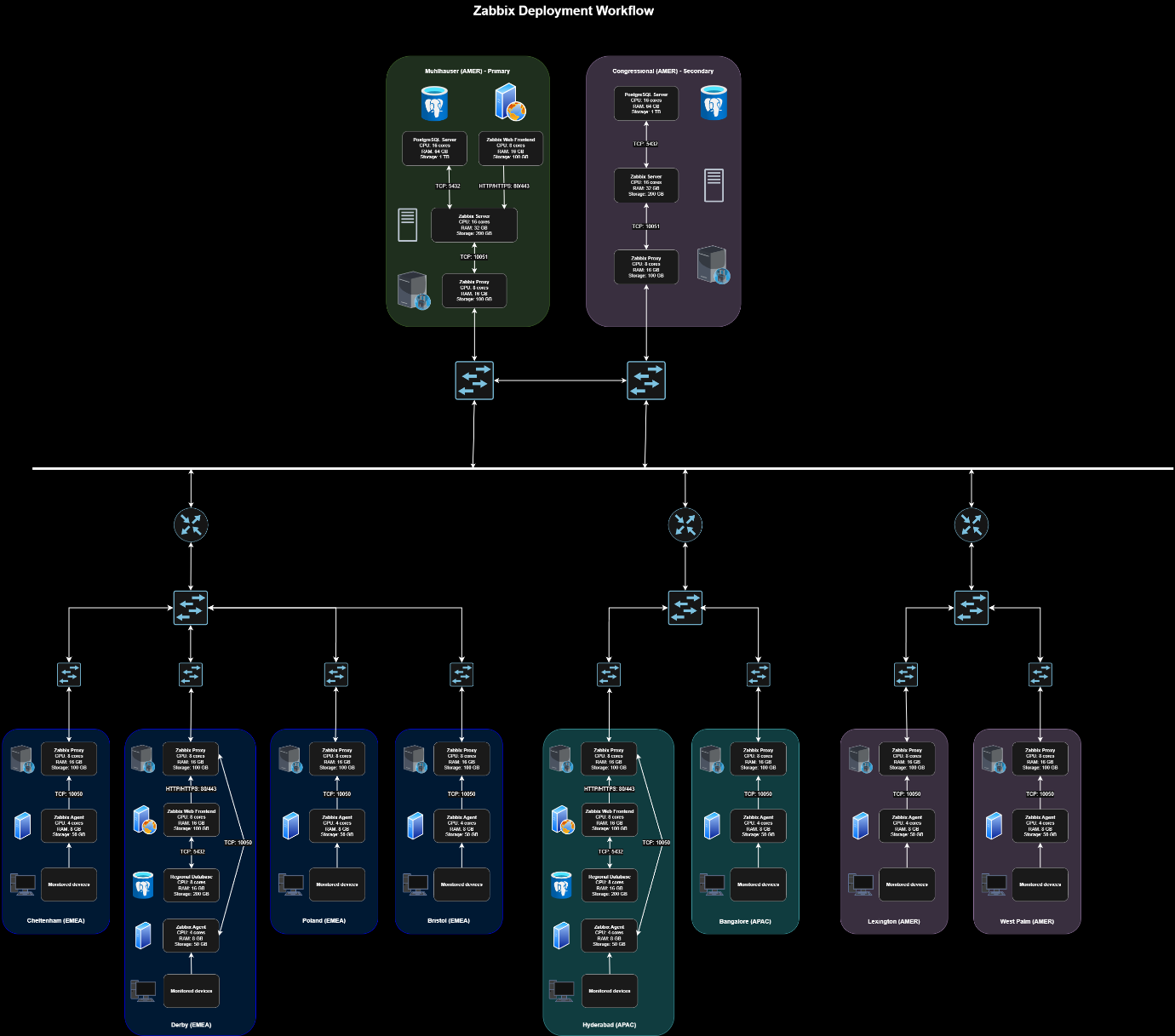
**Zabbix Deployment Workflow**



Below is a step-by-step installation guide tailored for a Zabbix deployment on Rocky Linux 9 based on the provided Zabbix Deployment Workflow diagram. This covers the Primary Zabbix Server, Secondary Zabbix Server, Proxies, and Agents across the specified regions.

**Step-by-Step Installation Guide**

**1. Prepare the Infrastructure**

* Ensure all servers (Primary Zabbix Server, Secondary Zabbix Server, Proxies, Agents) run Rocky Linux 9.
* Open required ports using firewalld.
  + TCP 10050 (proxy-agent communication)
  + TCP 10051 (server-agent communication)
  + TCP 80 or 443 (NGINX web frontend)

*sudo firewall-cmd --add-port=10050/tcp --permanent*

*sudo firewall-cmd --add-port=10051/tcp –permanent*

*sudo firewall-cmd --add-port=80/tcp –permanent*

*sudo firewall-cmd --add-port=443/tcp --permanent*

*sudo firewall-cmd –reload*

* Allocate storage
  + 100 GB for servers
  + 10 GB for proxies

**2. Install the Primary Zabbix Server**

* Install Dependencies: Update the system and install required packages.

*sudo dnf update*

*sudo dnf install epel-release*

*sudo dnf install nginx php php-fpm php-pgsql postgresql-server*

* Initialize and Start PostgreSQL.

*sudo postgresql-setup initdb*

*sudo systemctl enable postgresql*

*sudo systemctl start postgresql*

* Configure PostgreSQL:

*sudo -u postgres psql*

*CREATE DATABASE zabbix;*

*CREATE USER zabbix WITH PASSWORD 'your\_password';*

*GRANT ALL PRIVILEGES ON DATABASE zabbix TO zabbix;*

*\q*

* Add Zabbix Repository: Install the Zabbix 7.0 repository for Rocky Linux 9.

*sudo rpm -Uvh https://repo.zabbix.com/zabbix/7.0/rhel/9/x86\_64/zabbix-release-7.0-4.el9.noarch.rpm*

*sudo dnf clean all*

*sudo dnf makecache*

* Install Zabbix Server: Install server, frontend, and agent with PostgreSQL and NGINX support.

*sudo dnf install zabbix-server-pgsql zabbix-web-pgsql zabbix-nginx-conf zabbix-agent zabbix-sql-scripts*

* Import Schema: Import the initial schema into PostgreSQL.
* *sudo zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u zabbix psql zabbix*
* Configure Zabbix Server: Edit ‘/etc/zabbix/zabbix\_server.conf’ with database details.

*DBName=Zabbix*

*DBUser=Zabbix*

*DBPassword=your\_password*

* Configure PHP: Edit /etc/php.ini to set the timezone.

*date.timezone = UTC*

* Configure NGINX: The zabbix-nginx-conf package installs a default configuration file (/etc/nginx/conf.d/zabbix.conf). Verify or edit it to ensure proper settings.

*server {*

*listen 80;*

*server\_name <your\_server\_ip\_or\_domain>;*

*root /usr/share/zabbix;*

*index index.php;*

*location / {*

*try\_files $uri $uri/ /index.php?$args;*

*}*

*location ~ \.php$ {*

*include fastcgi\_params;*

*fastcgi\_pass unix:/var/run/php-fpm/www.sock;*

*fastcgi\_index index.php;*

*fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;*

*}*

*location ~\* \.(jpg|jpeg|png|gif|ico|css|js)$ {*

*access\_log off;*

*expires 30d;*

*}*

*}*

* Configure PHP-FPM: Ensure PHP-FPM is set to use a UNIX socket. Edit /etc/php-fpm.d/www.conf:

*listen = /var/run/php-fpm/www.sock*

*listen.owner = nginx*

*listen.group = nginx*

*listen.mode = 0660*

* Set Permissions: Adjust ownership for Zabbix and NGINX directories.

sudo chown -R zabbix:nginx /usr/share/zabbix

sudo chown -R nginx:nginx /var/lib/php/session

* Start Services:

*sudo systemctl enable nginx php-fpm zabbix-server zabbix-agent*

*sudo systemctl start nginx php-fpm zabbix-server zabbix-agent*

**3. Install the Secondary Zabbix Server**

* Repeat the Primary Server steps on the secondary server, including NGINX and PHP-FPM setup.
* Set up PostgreSQL replication.
* Update ‘/etc/zabbix/zabbix\_server.conf’ to sync with the primary server.

*HANodeName=zbx-secondary*

*NodeAddress=<secondary\_server\_ip>:10051*

* Ensure NGINX configuration matches the primary server.

**4. Install Zabbix Proxies**

* Install proxies for all regions
* Install Proxy:

*sudo dnf install zabbix-proxy-pgsql*

* Configure Database: Set up a local PostgreSQL database for each proxy.

*sudo -u postgres psql*

*CREATE DATABASE zabbix\_proxy\_<region>;*

*CREATE USER zabbix\_proxy\_<region> WITH PASSWORD 'your\_password';*

*GRANT ALL PRIVILEGES ON DATABASE zabbix\_proxy\_<region> TO zabbix\_proxy\_<region>;*

*\q*

* Import schema:

*sudo zcat /usr/share/zabbix-sql-scripts/postgresql/schema.sql.gz | sudo -u zabbix\_proxy\_<region> psql zabbix\_proxy\_<region>*

* Configure Proxy: Edit ‘/etc/zabbix/zabbix\_proxy.conf’ with proxy name, server IP, and database details.

*Server=<primary\_server\_ip>*

*DBName=zabbix\_proxy\_<region>*

*DBUser=zabbix\_proxy\_<region>*

*DBPassword=your\_password*

*ProxyMode=0 # Active proxy*

* Start Proxy:

*sudo systemctl enable zabbix-proxy*

*sudo systemctl start zabbix-proxy*

**5. Install Zabbix Agents**

* Install agents on all monitored devices under each proxy.
* Install Agent:

*sudo dnf install zabbix-agent*

* Configure Agent: Edit `/etc/zabbix/zabbix\_agentd.conf` to point to the respective proxy IP.

*Server=<proxy\_ip>*

*ServerActive=<proxy\_ip>*

*Hostname=<device\_name>*

* Start Agent:

*sudo systemctl enable zabbix-agent*

*sudo systemctl start zabbix-agent*

**6. Configure Communication**

* Verify connectivity between primary server, secondary server, proxies, and agents using ‘zabbix\_get’ or ‘netstat’.
* Add proxies and monitored hosts in the Zabbix web interface (‘http://<server\_ip>/zabbix’).

**Configuration > Proxies > Create Proxy**: Add each proxy.

**Configuration > Hosts > Create Host**: Add devices with proxy assignments.

**7. Test and Validate**

* Log in to the Zabbix frontend (default: Admin/zabbix).
* Check statuses in **Monitoring > Hosts** and **Monitoring > Proxies**.
* Verify data collection from all regions in **Monitoring > Latest Data**.

**8. Finalize Deployment**

* Configure monitoring templates, triggers, and alerts via the web interface.
* Back up PostgreSQL databases and configuration files (/etc/zabbix/, /etc/nginx/).
* Secure the setup:
  + Change default passwords
  + Enable HTTPS by configuring NGINX with SSL/TLS
  + Restrict database access by updating pg\_hba.conf.
  + Secure NGINX by disabling directory listing and restricting access to sensitive files.
* Example NGINX SSL configuration (after obtaining certificates):

*server {*

*listen 443 ssl;*

*server\_name <your\_server\_ip\_or\_domain>;*

*ssl\_certificate /etc/letsencrypt/live/<your\_domain>/fullchain.pem;*

*ssl\_certificate\_key /etc/letsencrypt/live/<your\_domain>/privkey.pem;*

*root /usr/share/zabbix;*

*location / {*

*try\_files $uri $uri/ /index.php?$args;*

*}*

*location ~ \.php$ {*

*include fastcgi\_params;*

*fastcgi\_pass unix:/var/run/php-fpm/www.sock;*

*fastcgi\_index index.php;*

*fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;*

*}*

*}*

**PostgreSQL Replication for Secondary Server**

Setting up PostgreSQL replication to configure the Secondary Zabbix Server as a standby server involves configuring streaming replication with a hot standby setup. This ensures high availability (HA) for the Zabbix database, aligning with the Zabbix HA documentation recommendations. Below are detailed steps for Rocky Linux 9 using PostgreSQL, based on the Zabbix Deployment Workflow.

**Prerequisites**

* Two servers: Primary Zabbix Server (master) and Secondary Zabbix Server (standby).
* Both servers have PostgreSQL installed and the Zabbix database (zabbix) created (as per the previous steps).
* Network connectivity between servers with ports 5432 (PostgreSQL) and 22 (SSH) open.
* rsync and ssh configured for file transfer between servers.
* root or sudo access on both servers.

**Step-by-Step Guide to Configure PostgreSQL Replication**

1. Prepare the Primary Server

* Stop the Primary Server: Ensure no changes occur during the initial setup.

*sudo systemctl stop postgresql*

* Backup the Data Directory: Create a backup of the current PostgreSQL data directory

*sudo cp -r /var/lib/pgsql/15/data/ /var/lib/pgsql/15/data\_backup/*

* Configure the primary server for replication.

Edit /var/lib/pgsql/15/data/postgresql.conf and set the following parameters:

*wal\_level = replica*

*max\_wal\_senders = 10*

*wal\_keep\_size = 128MB*

*hot\_standby = off # Enable on standby later*

* Allow replication connections from the standby server.

Edit /var/lib/pgsql/15/data/pg\_hba.conf & add the following line (replace `<standby\_ip>` with the standby server's IP):

*host replication zabbix\_repl <standby\_ip>/32 md5*

* Restart PostgreSQL to apply the changes.

*sudo systemctl start postgresql*

* Create a role for replication.

*sudo -u postgres psql*

*CREATE ROLE zabbix\_repl WITH REPLICATION LOGIN PASSWORD 'your\_repl\_password';*

*\q*

2. Prepare the Standby Server

* Stop PostgreSQL: Ensure the standby server is stopped.

*sudo systemctl stop postgresql*

* Remove Existing Data: Clear the existing data directory to avoid conflicts.

*sudo rm -rf /var/lib/pgsql/15/data/\**

* Sync Data from Primary: Use rsync to copy the primary’s data directory to the standby. On the standby server, run:

*sudo -u postgres rsync -av -e ssh postgres@<primary\_ip>:/var/lib/pgsql/15/data/\* /var/lib/pgsql/15/data/*

* Ensure SSH is set up with key-based authentication for automation. Configure the standby: Edit /var/lib/pgsql/15/data/postgresql.conf

*hot\_standby = on*

* Configure replication in postgresql.conf:

*standby\_mode = 'on'*

*primary\_conninfo = 'host=<primary\_ip> port=5432 user=zabbix\_repl password=your\_repl\_password'*

*trigger\_file = '/tmp/postgresql.trigger'*

* Set Permissions: Ensure the data directory is owned by the ‘postgres’ user.

*sudo chown -R postgres:postgres /var/lib/pgsql/15/data*

3. Start the Standby Server

* Start the PostgreSQL service on the standby server.

*sudo systemctl start postgresql*

* Verify recovery mode:

*sudo -u postgres psql -c "SELECT pg\_is\_in\_recovery();"*

It should return ‘t’ (true) if in recovery mode.

4. Test and Monitor Replication

* Check Replication Status: On the primary, run:

*sudo -u postgres psql -c "SELECT \* FROM pg\_stat\_replication;"*

This should show the standby server’s connection.

* Test Failover: Stop the primary server and confirm the standby takes ove. On standby, promote it to primary:

*sudo -u postgres touch /tmp/postgresql.trigger*

*sudo systemctl restart postgresql*

* Reconfigure if Needed: After failover, set up the old primary as the new standby by reversing the process.

5. Integrate with Zabbix HA

* Configure Zabbix HA in /etc/zabbix/zabbix\_server.conf on both servers:

*# Primary*

*HANodeName=zbx-primary*

*NodeAddress=<primary\_ip>:10051*

*# Secondary*

*HANodeName=zbx-secondary*

*NodeAddress=<secondary\_ip>:10051*

* Enable HA in the Zabbix frontend (**Administration > General > HA**).
* Test failover by stopping the primary Zabbix server and verifying the secondary takes over.

6. Finalize and Secure

* Enable automatic startup on both servers:

*sudo systemctl enable postgresql*

* Regularly back up the PostgreSQL WAL files and configuration.

Zabbix HA Documentation <https://www.zabbix.com/documentation/current/en/manual/installation/high_availability>

PostgreSQL Replication Docs

<https://www.postgresql.org/docs/current/high-availability.html>