**Zabbix Minimal UAT Setup**

This minimal setup provides a functional Zabbix UAT environment with reduced complexity and resources, suitable for testing core features before production deployment. Below is a step-by-step guide for a minimal UAT setup on Rocky Linux 9 using Zabbix 7.0 LTS, NGINX, and PostgreSQL, without HA.

**Architecture:**

* **Single Server:** Hosts the PostgreSQL database, Zabbix server, and NGINX web frontend to reduce complexity and resource needs.
* **Zabbix Proxy Server(s):** One or more dedicated proxy servers to simulate regional monitoring (optional).
* **Zabbix Agents:** Installed on test devices to simulate monitored hosts.

**Step-by-Step Installation Guide**

**1. Prepare the Infrastructure**

* Minimal Resource Recommendations
  + Main Server (PostgreSQL, Zabbix Server, NGINX):
    - CPU: 4 cores
    - RAM: 8 GB
    - Storage: 100 GB
  + Zabbix Proxy Server
    - CPU: 2 cores
    - RAM: 4 GB
    - Storage: 50 GB
* Open required ports using firewalld:

PostgreSQL Server: TCP 5432 (local access only)

Zabbix Server: TCP 10051 (proxy/agent communication)

NGINX: TCP 80 (443 if HTTPS is configured)

Zabbix Proxy Server: TCP 10050 (agent communication)

*sudo firewall-cmd --add-port=5432/tcp –permanent*

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

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

*sudo firewall-cmd --reload*

* Ensure the main server is reachable by proxies (if used) and agents on ports 10051 and 80.

**2. Install the PostgreSQL on the Main Server**

* **Install PostgreSQL**

*sudo dnf install postgresql-server*

* **Verify PostgreSQL Version**

*psql --version*

This guide assumes PostgreSQL 15

* **Initialize and Start PostgreSQL**

*sudo postgresql-setup initdb*

*sudo systemctl enable postgresql*

*sudo systemctl start postgresql*

* **Configure PostgreSQL**

Create the Zabbix database and user.

*sudo -u postgres psql*

*CREATE DATABASE zabbix;*

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

*GRANT ALL PRIVILEGES ON DATABASE zabbix TO zabbix;*

*\q*

* **Allow Local Connections**

No remote access required unless proxies use a remote DB

Edit /var/lib/pgsql/15/data/pg\_hba.conf

*host zabbix zabbix 127.0..0.1/32 md5*

* **Restart PostgreSQL**

*sudo systemctl restart postgresql*

**3. Install Zabbix Server on the Main Server**

* **Install Dependencies**

*sudo dnf install epel-release*

* **Add Zabbix Repository**

*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 and Agent**

*sudo dnf install zabbix-server-pgsql zabbix-agent zabbix-sql-scripts*

* **Import Schema**

*sudo zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | psql -h 127.0.0.1 -U zabbix -d zabbix*

Enter the zabbix password when prompted.

* **Configure Zabbix Server**

Edit /etc/zabbix/zabbix\_server.conf

*DBHost=127.0.0.1*

*DBName=zabbix*

*DBUser=zabbix*

*DBPassword=zabbix\_password*

* **Start Services**

*sudo systemctl enable zabbix-server zabbix-agent*

*sudo systemctl start zabbix-server zabbix-agent*

**4. Install NGINX Web Frontend on the Main Server**

* **Install Dependencies**

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

* **Install Zabbix Web Frontend**

*sudo dnf install zabbix-web-pgsql zabbix-nginx-conf*

* **Configure PHP**

Edit /etc/php.ini

*date.timezone = UTC*

* **Configure NGINX**

Edit /etc/nginx/conf.d/zabbix.conf

*server {*

*listen 80;*

*server\_name localhost;*

*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;*

*}*

*}*

* **Configure PHP-FPM**

Edit /etc/php-fpm.d/www.conf

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

*listen.owner = nginx*

*listen.group = nginx*

*listen.mode = 0660*

* **Configure Zabbix Frontend**

Edit /etc/zabbix/web/zabbix.conf.php:

*$DB['TYPE'] = 'POSTGRESQL';*

*$DB['SERVER'] = '127.0.0.1';*

*$DB['PORT'] = '5432';*

*$DB['DATABASE'] = 'zabbix';*

*$DB['USER'] = 'zabbix';*

*$DB['PASSWORD'] = 'zabbix\_password';*

*$ZBX\_SERVER = '127.0.0.1';*

*$ZBX\_SERVER\_PORT = '10051';*

*$ZBX\_SERVER\_NAME = 'Zabbix UAT';*

* **Set Permissions**

*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*

*sudo systemctl start nginx php-fpm*

**5. Install Zabbix Proxy (Optional)**

* **Update system & Open ports**

*sudo dnf update*

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

*sudo firewall-cmd --reload*

* **Install Proxy**

*sudo dnf install epel-release*

*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*

*sudo dnf install zabbix-proxy-pgsql postgresql-server*

* **Initialize and Start PostgreSQL (local to the proxy server):**

*sudo postgresql-setup initdb*

*sudo systemctl enable postgresql*

*sudo systemctl start postgresql*

* **Configure Database**

*sudo -u postgres psql*

*CREATE DATABASE zabbix\_proxy\_uat;*

*CREATE USER zabbix\_proxy\_uat WITH PASSWORD 'proxy\_password';*

*GRANT ALL PRIVILEGES ON DATABASE zabbix\_proxy\_uat TO zabbix\_proxy\_uat;*

*\q*

* **Import Schema**

*sudo zcat /usr/share/zabbix-sql-scripts/postgresql/schema.sql.gz | psql -U zabbix\_proxy\_uat -d zabbix\_proxy\_uat*

* **Configure Proxy**

Edit /etc/zabbix/zabbix\_proxy.conf:

*Server=<main\_server\_ip>*

*DBName=zabbix\_proxy\_uat*

*DBUser=zabbix\_proxy\_uat*

*DBPassword=proxy\_password*

*ProxyMode=0 # Active proxy*

*Hostname=Zabbix Proxy UAT*

* **Start Proxy**

*sudo systemctl enable zabbix-proxy postgresql*

*sudo systemctl start zabbix-proxy postgresql*

**6. Install Zabbix Agents**

Install agents on all test devices

* **Install Agent**

*sudo dnf install zabbix-agent*

* **Configure Agent**

Edit /etc/zabbix/zabbix\_agentd.conf

*Server=<proxy\_server\_ip or main\_server\_ip>*

*ServerActive=<proxy\_server\_ip or main\_server\_ip>*

*Hostname=<device\_name>*

Use the proxy IP if testing with a proxy; otherwise, use the main server IP.

* **Start Agent**

*sudo systemctl enable zabbix-agent*

*sudo systemctl start zabbix-agent*

**7. Configure Communication**

**Verify connectivity:**

* Zabbix Server to PostgreSQL Server (local, port 5432).
* Proxy (if used) to Zabbix Server (port 10051).
* Agents to Proxy (port 10050) or Zabbix Server (port 10051).

*nc -zv <main\_server\_ip> 10051*

**Add proxy and hosts:**

* Access the Zabbix web interface using default login Admin/zabbix (http://<main\_server\_ip>/zabbix)
* If using a Proxy:

Configuration > Proxies > Create Proxy: Add the proxy with its IP and name.

* Add test hosts:

Configuration > Hosts > Create Host: Add devices with proxy assignments (if used) or direct to the Zabbix server.

**8. Test and Validate**

* Log in to the Zabbix frontend (http://<nginx\_server\_ip>/zabbix)
* Check status:

Monitoring > Hosts: Verify test hosts are listed.

Monitoring > Proxies: Confirm proxy status.

Monitoring > Latest Data: Check data collection from the agents.

* Test Alerts: Configure a basic trigger (e.g., CPU load) and verify notifications (via email or internal messages).

**9. Finalize UAT Setup**

* Basic Backup:
  + Back up the database:

*pg\_dump -U zabbix zabbix > /path/to/backup/zabbix\_backup.sql*

* + Backup configs:

*tar -czf /path/to/backup/zabbix\_configs.tar.gz /etc/zabbix/ /etc/nginx/ /usr/share/zabbix/*

* Monitoring Templates: Apply the default templates (e.g., Linux by Zabbix Agent) to test hosts.
* Minimal Security:
  + Change the default Admin/zabbix password in the web interface.
  + Optionally enable HTTPS on NGINX for testing (use a self-signed certificate).

**Notes for UAT**

**1. Skipping Proxies:**

If regional monitoring is not critical for UAT, skip the proxy server and have agents connect directly to the Zabbix server (`Server=<main\_server\_ip>` in `zabbix\_agentd.conf`).

**2. Resource Scaling:**

The recommended 4 cores/8 GB RAM for the main server is suitable for small-scale testing. Increase to 8 cores/16 GB if testing with more hosts or heavier workloads.

**3. Security:**

PSK and SSL are omitted for simplicity. If needed for testing, follow the setup mentioned in the document.

**4. Failover:**

No HA is configured, so no failover testing is required. If failover is needed for UAT, consider adding a second Zabbix server but skip PostgreSQL replication.

**5. Testing Scope:**

Focus on core functionality: host monitoring, data collection, triggers, and web interface usability. If proxies are included, test proxy-to-server communication and regional data aggregation.

**References**

Zabbix 7.0 Documentation: <https://www.zabbix.com/documentation/7.0/>

PostgreSQL Documentation: <https://www.postgresql.org/docs/15/>

NGINX Configuration: <https://www.nginx.com/resources/wiki/>

**Suggested Additions for testing:**

1. **Testing Production-Like Configurations**:

Follow these steps for enabling PSK and HTTPS to test production-like security:

**PSK for Proxy/Agent Communication**:

On proxy or agent, generate PSK

*sudo openssl rand -hex 32 > /etc/zabbix/zabbix\_proxy.psk*

Edit /etc/zabbix/zabbix\_proxy.conf or zabbix\_agentd.conf

*TLSConnect=psk*

*TLSAccept=psk*

*TLSPSKIdentity=ZabbixProxy\_UAT*

*TLSPSKFile=/etc/zabbix/zabbix\_proxy.psk*

On Zabbix server, edit /etc/zabbix/zabbix\_server.conf

*TLSConnect=psk*

*TLSAccept=psk*

Configure PSK in web interface: Configuration > Proxies > Encryption

**HTTPS for NGINX** (using a self-signed certificate for testing):

Generate self-signed certificate

*sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt*

Update /etc/nginx/conf.d/zabbix.conf

*server {*

*listen 443 ssl;*

*server\_name localhost;*

*ssl\_certificate /etc/nginx/ssl/nginx.crt;*

*ssl\_certificate\_key /etc/nginx/ssl/nginx.key;*

*root /usr/share/zabbix;*

*...*

*}*

Open port 443

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

*sudo firewall-cmd --reload*

Restart NGINX

*sudo systemctl restart nginx*

1. **Monitoring Templates and Triggers**:

Configure a sample template and trigger. Go to Zabbix web interface:

**Configuration > Hosts**, select a test host, and link the “Linux by Zabbix Agent” template.

Create a trigger: **Configuration > Triggers > Create Trigger**:

* + Name: “High CPU Load”
  + Expression: {<host>:system.cpu.load[all,avg1].last()}>5
  + Severity: Warning

Create an action: **Configuration > Actions > Trigger Actions > Create Action**:

* + Condition: Trigger severity >= Warning
  + Operation: Send message to user “Admin” via internal notification.

1. **Integration Testing**:

Configure and test a basic notification integration (e.g., email).

Configure email media type: **Administration > Media Types > Email**:

* + SMTP server: <test\_smtp\_server>
  + SMTP email: zabbix@test.com
  + Test with a SMTP server (e.g., a local Postfix or external service).

Assign email media to the Admin user: **Administration > Users > Admin > Media**. Trigger a test alert to verify email delivery.

1. **Performance Testing**:

Simulate production-like load:

* + Deploy 20–50 test agents on VMs/containers to mimic production host count.
  + Use a script to generate synthetic data (e.g., via Zabbix sender):

Install zabbix-sender

*sudo dnf install zabbix-sender*

Send test data

*zabbix\_sender -z <main\_server\_ip> -p 10051 -s "<host>" -k "test.metric" -o 42*

Monitor server performance: **Monitoring > Hosts > Zabbix Server > Graphs** (use “Zabbix Server” template metrics like CPU/memory).

1. **Backup and Restore Testing**:

The backup steps are already included so test restore following these steps:

* + Restore database:

Drop and recreate database

*sudo -u postgres psql -c "DROP DATABASE zabbix;"*

*sudo -u postgres psql -c "CREATE DATABASE zabbix;"*

*sudo -u postgres psql -c "GRANT ALL PRIVILEGES ON DATABASE zabbix TO zabbix;"*

Restore

*psql -h 127.0.0.1 -U zabbix -d zabbix < /path/to/backup/zabbix\_backup.sql*

* + Restart services:

*sudo systemctl restart zabbix-server nginx php-fpm*

* + Verify data in the web interface.