






# **Занятие 1: Введение в мониторинг + Zabbix**

 **Цель:** Понять концепцию мониторинга и научиться собирать метрики в Zabbix

## Что такое мониторинг





Мониторинг — это сбор, хранение и анализ данных о состоянии систем.

Позволяет:

-  Предупреждать инциденты
-  Находить узкие места
-  Контролировать SLA
-  Повышать надежность инфраструктуры

 *Если не измеряешь — не можешь улучшить.*

## Зачем нужен мониторинг

-  Своевременное обнаружение проблем
-  Планирование ресурсов (нагрузка, производительность)
-  Аналитика и оптимизация
-  Алертинг — уведомления о критических состояниях

## Типы мониторинга

Тип	Что контролируем	Примеры
Инфраструктурный	CPU, RAM, диск, сеть	Zabbix, Prometheus
Прикладной	API, веб-сервисы, процессы	Grafana, NewRelic
Бизнес-мониторинг	Кол-во заказов, транзакций	ELK, BI-системы
Логирование	Ошибки, события	ELK, Loki

## **Логирование и алертинг**

- **Логирование** — запись событий (ошибки, запросы, действия)
- **Алертинг** — уведомления о проблемах (email, Telegram, Slack)

 **Цель:** чтобы DevOps узнал о проблеме **раньше пользователей.**






## Обзор инструментов мониторинга

Инструмент	Особенности
Zabbix	Классический, универсальный мониторинг
Prometheus	Cloud-native, time-series, интеграция с Kubernetes
Grafana	Визуализация данных из разных источников
ELK	Аналитика логов (Elasticsearch + Logstash + Kibana)

В этом курсе начнем с **Zabbix** — понятный и визуальный инструмент.

# Архитектура Zabbix

## Компоненты:

-  **Server** — ядро системы, собирает и обрабатывает данные
-  **Agent** — собирает метрики с хостов
-  **Proxy** — промежуточный узел (для распределённых систем)
-  **Database** — хранит данные
-  **Frontend** — веб-интерфейс

 *Zabbix Server* ↔ *Agent* ↔ *DB* ↔ *Web UI*

## Практика: установка Zabbix

**Задача:** развернуть Zabbix через Docker Compose (или вручную в Ubuntu)

**Через Docker Compose:**

```
git clone https://github.com/zabbix/zabbix-docker.git
cd zabbix-docker
docker compose up -d
```

 После запуска открыть в браузере:

```
http://localhost:8080
```



## Установка Zabbix Agent

На локальной VM (Ubuntu):

```
sudo apt update  
sudo apt install zabbix-agent -y
```

Добавить IP сервера Zabbix в `/etc/zabbix/zabbix_agentd.conf` :

```
Server=192.168.x.x  
ServerActive=192.168.x.x
```

Проверить работу:

```
sudo systemctl status zabbix-agent
```



## Настройка хоста и шаблона

1. Configuration → Hosts → Create Host
2. Указать имя и IP
3. Привязать шаблон **Linux OS by Zabbix agent**
4. Проверить статус (зелёный значок = всё ок)

Через несколько минут появятся метрики CPU, RAM, Disk, Network.



## Просмотр графиков и триггеров

- *Monitoring* → *Latest Data* — последние значения
- *Monitoring* → *Graphs* — визуализация
- *Triggers* — события по условиям (например, CPU > 80%)



Пример триггера:

```
{Template OS Linux:system.cpu.util.last()} > 80
```

## Настройка алертов

Пример: уведомление при CPU > 80%

1. Создать **Trigger**
2. Создать **Media type** — Email или Telegram
3. Настроить **Action** — кого уведомлять и при каких условиях

 Telegram — удобен для учебных целей (бот + API key)

## Домашнее задание

- ✓ Добавить второй хост
- ✓ Настроить мониторинг доступности сервиса ( `ping` или `http` )
- ★ (по желанию) — добавить уведомление через **email** или **Telegram**

## Резюме

Сегодня вы узнали:

- Что такое мониторинг и зачем он нужен
- Основные типы мониторинга
- Как устроен Zabbix
- Как собрать метрики и настроить алерты

... На следующем занятии: Prometheus + NodeExporter.