Portfolio DevOps & Infrastructure Automation

Implementasi End-to-End CI/CD, Cloud Infrastructure, dan Monitoring System

Project Name: DumbMerch E-Commerce Platform

Candidate: Fauzan

Project Overview & Tech Stack

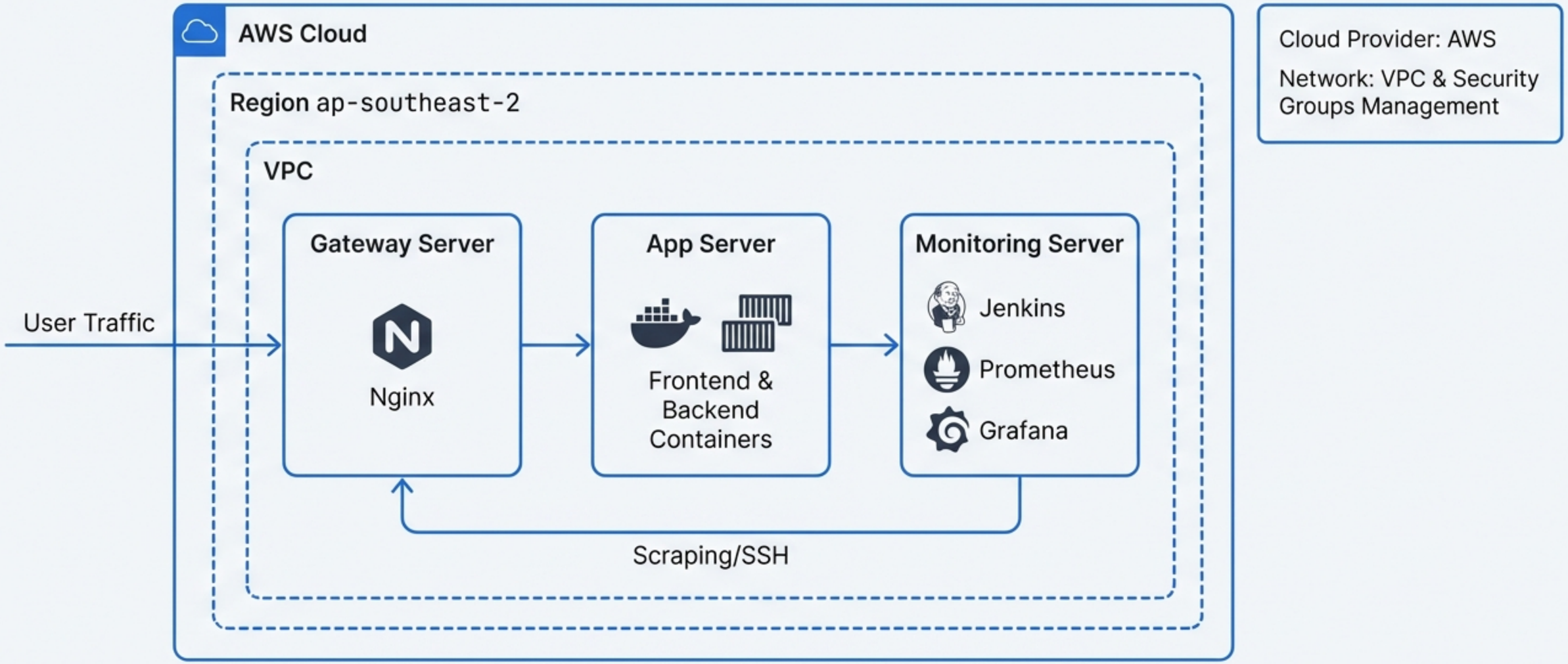
Aplikasi Microservices

- Frontend: `React.js` (Node.js environment)
- Backend: `Golang` (Go v1.16)
- Database: `PostgreSQL` (Containerized)

Tools Infrastruktur

- Cloud: `AWS` (EC2 Region ap-southeast-2)
- IaC & Config: `Terraform`, `Ansible`
- CI/CD: `Jenkins`, `Docker`, `SonarQube`
- Observability: `Prometheus`, `Grafana`, `Alertmanager`

Arsitektur Sistem & Topologi Cloud



Provisioning Infrastruktur (Terraform)

Automated EC2 Creation:
Menggunakan resource `aws_instance` untuk deployment server utama dan monitoring.

Dynamic Security Groups:
Konfigurasi inbound/outbound rules untuk SSH (22), HTTP (80), dan App Ports.

Output Management:
Auto-generate Public IP untuk akses server instan.

```
resource "aws_instance" "server" {
  ami           = "ami-0e7a5785ab62399cb"
  instance_type = "c7i-flex.large"
  key_name      = "keyaws"

  tags = {
    Name = "terraform-server"
  }

  vpc_security_group_ids = [
    aws_security_group.server_sg.id
  ]
}
```

```
resource "aws_security_group" "allow_ssh_http" {
  name = "allow-ssh-http"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Server Configuration Management (Ansible)

Inventory Management

```
[appserver]
103.103.23.208
13.236.136.137

[monitoring]
3.107.29.73

[gateway]
103.31.205.58

[all:vars]
ansible_user=devops
ansible_python_interpreter=/usr/bin/python3
```

User & Security

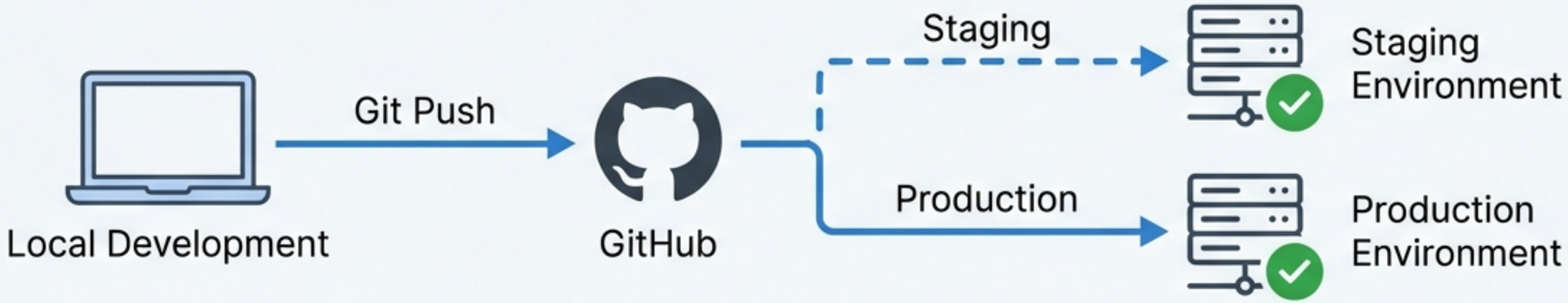
Setup user `devops` dengan akses sudo dan SSH Key setup.

Docker Automation

```
- name: Install required packages
  apt:
    name:
      - docker-ce
      - docker-ce-cli
    state: present
```

Instalasi Docker Engine dan permissions user group.

Workflow Repository & Branching Strategy



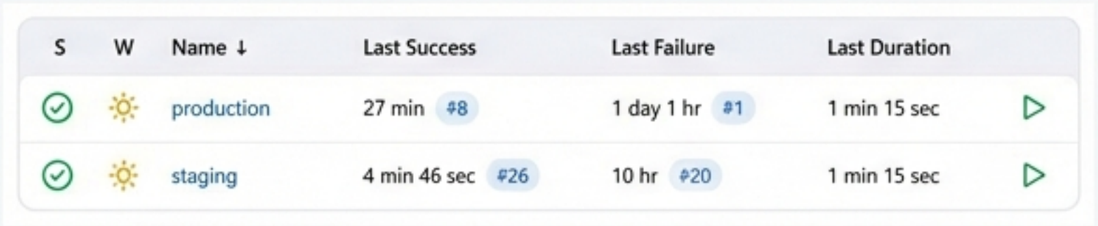
- **Environment Strategy**: Pemisahan branch `staging` dan `production`.
- **Git Automation**: Ansible playbook untuk setup identitas Git dan cloning repository privat.
- **Sync Mechanism**: Skrip otomatis sinkronisasi kode antar environment.

```yaml
- name: Create and push branches (backend)
  shell: |
    git checkout -b staging || git checkout staging
    git push -u origin staging || true
    git checkout -b production || git checkout production
    git push -u origin production || true
  args:
    chdir: "{{ app_dir }}/backend"
    become_user: "{{ ansible_user }}"
```

```yaml
- name: Create and push branches (frontend)
  shell: |
    git checkout -b staging || git checkout staging
    git push -u origin staging || true
    git checkout -b production || git checkout production
    git push -u origin production || true
  args:
    chdir: "{{ app_dir }}/frontend"
    become_user: "{{ ansible_user }}"
```

Continuous Integration (Jenkins & SonarQube)

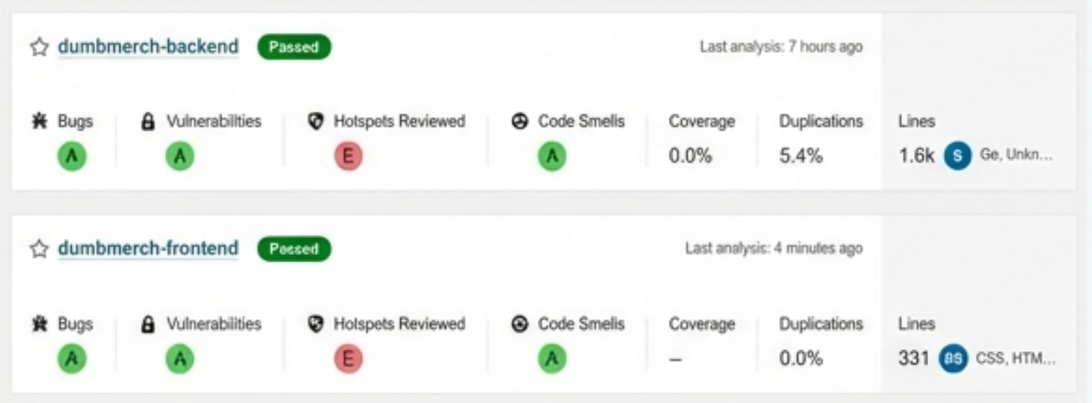
Pipeline Automation



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	production	27 min #8	1 day 1 hr #1	1 min 15 sec
✓	☀	staging	4 min 46 sec #26	10 hr #20	1 min 15 sec

Jenkins Multibranch Pipeline dengan webhook trigger dari GitHub.

Quality Assurance

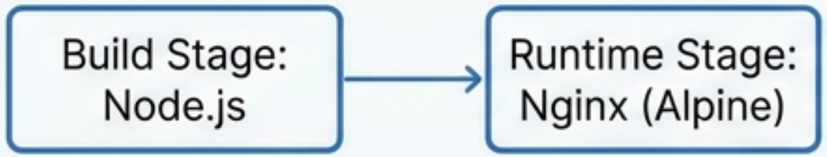


Integrasi SonarQube Quality Gate: Analisis bugs, vulnerabilities, dan code smells sebelum build.

Strategi Containerization (Docker Multi-stage Build)

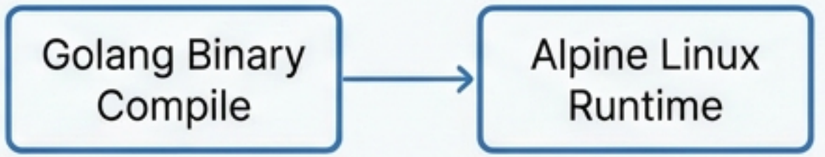
Frontend Optimization

```
#BUILD STAGE
FROM node:16-alpine AS build
WORKDIR /app
COPY package*.json ./
RUN npm ci

COPY . .
RUN npm run build


#RUNTIME STAGE
FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```



Backend Optimization

```
FROM golang:1.16-alpine AS builder
WORKDIR /app
RUN apk add --no-cache git
COPY go.mod go.sum ./
RUN go mod download
COPY . .
RUN CGO_ENABLED=0 go build -o app

FROM alpine:3.18
WDRKDIR /app
RUN apk add --no-cache ca-certificates
# Copy binary dari builder
COPY --from=builder /app/app .
# Copy .env file ke dalam container
COPY --from=builder /app/.env .env
# Expose port
EXPOSE 8080
CMD ["./app"]
```



Efficiency: Image ringan dan aman untuk deployment production.

Continuous Deployment (CD) Automation Flow

Automated Flow: Pull -> Build -> Stop Old -> Run New -> Health Check

Deployment Backend & Database Integration

.env Configuration

```
SECRET_KEY=bolehapaaja
PATH_FILE=http://13.236.136.137:5000/uploads/
SERVER_KEY=SB-Mid-server-fJAy6udMPnJCIyFguce8Eot3
CLIENT_KEY=SB-Mid-client-YUogx3u74Gq9MTMS
EMAIL_SYSTEM=demo.dumbways@gmail.com
PASSWORD_SYSTEM=rbgmgzzcmrfdtbpu

DB_HOST=postgres
DB_USER=dumbmerch
DB_PASSWORD=devops123
DB_NAME=dumbmerch
DB_PORT=5432
PORT=8080
```

Manajemen environment variables untuk kredensial DB dan payment gateway.

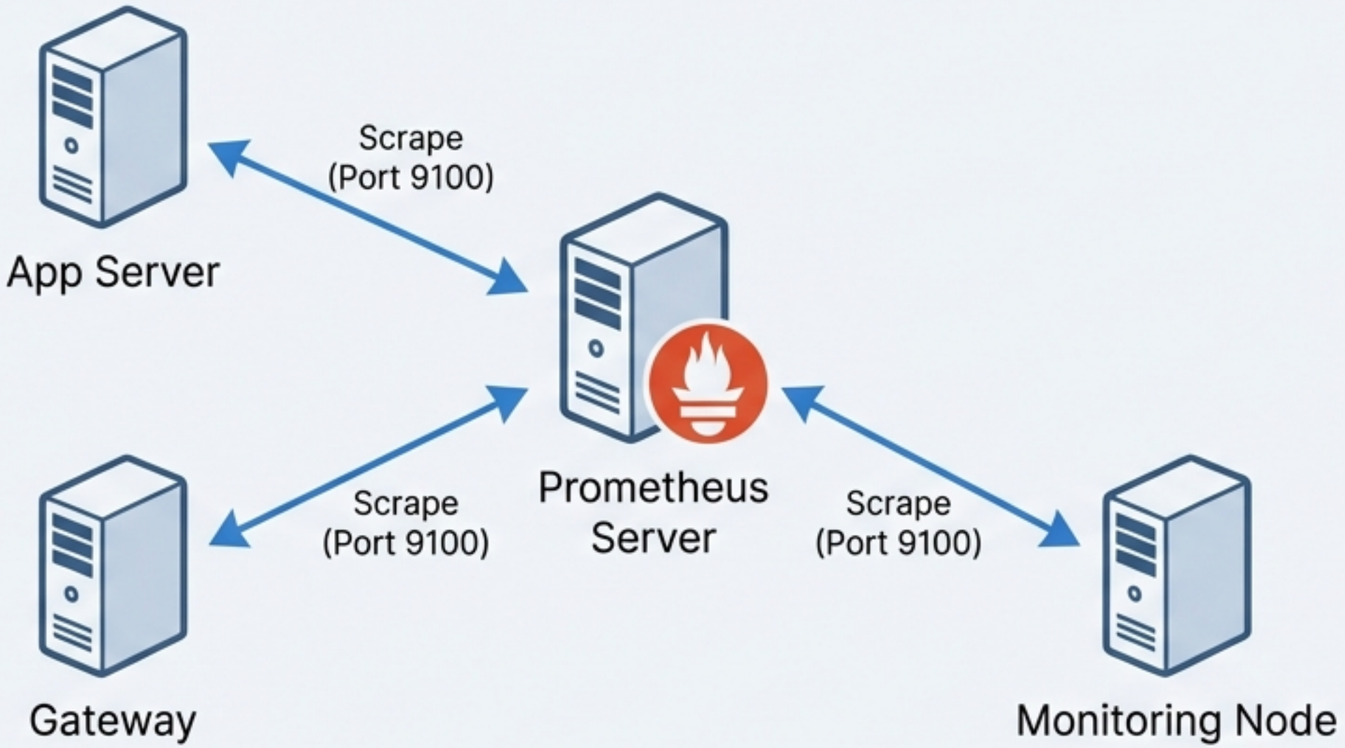
Database Verification

```
deveps@ip-172-31-14-165:~$ docker exec -it postgres psql -U dumbmerch -d dumbmerch
psql (15.15)
Type "help" for help.

dumbmerch=# \dt
                List of relations
 Schema |        Name         | Type  |   Owner
--------+---------------------+-------+-----------
 public | cetegeries          | table | dumbmerch
 public | preduct_categories  | table | dumbmerch
 public | products            | table | dumbmerch
 public | profiles            | table | dumbmerch
 public | transactions        | table | dumbmerch
 public | users               | table | dumbmerch
(6 rows)

dumbmerch=# SELECT * FROM users;
 id |    name    |     email      |                          password                          |  status  |         created_at         |         updated_at
----+------------+----------------+------------------------------------------------------------+----------+----------------------------+----------------------------
  1 | Admin Test | admin@test.com | password123                                                | active   | 2026-01-17 18:19:59.426182+00 | 2026-01-17 18:19:59.426182+00
  3 | Test       | test@test.com  | $2a$10$EH5oxgQIaOpSIGuBC21.dTeOafVrGR9.h3QjCHBwLZ4kq | customer | 2026-01-17 18:42:00.031499+00 | 2026-01-17 18:42:00.031499+00
(2 rows)
```

Validasi koneksi dan skema database via container PostgreSQL.

Sistem Monitoring (Prometheus & Node Exporter)

The Scraping Architecture



Prometheus sentralisasi pengumpulan metrik CPU, RAM, dan Disk dari seluruh node.

Ansible Task: Install Node Exporter

```
- name: "run docker compose"
  community.docker.docker_compose_v2:
    files:
      - node-exporter.yml
    project_name: monitoring
    project_src: "/home/devops/monitoring"
```

Docker Compose Definition

```
services:
  node_exporter:
    image: prom/node-exporter:latest
    container_name: node_exporter
    restart: always
    command:
      - "--web.listen-address=:9100"
      - "--path.procfs=/host/proc"
      - "--path.sysfs=/host/sys"
      - "--path.rootfs=/rootfs"
    ports:
      - "9100:9100"
```

Visualisasi Data & Dashboard (Grafana)

Integration:
Prometheus
Data Source

Real-time:
Visualisasi performa infrastruktur live

Accessibility:
URL publik dengan autentikasi dasar

Alerting System & Incident Response

Problem Definition: Alert Rules

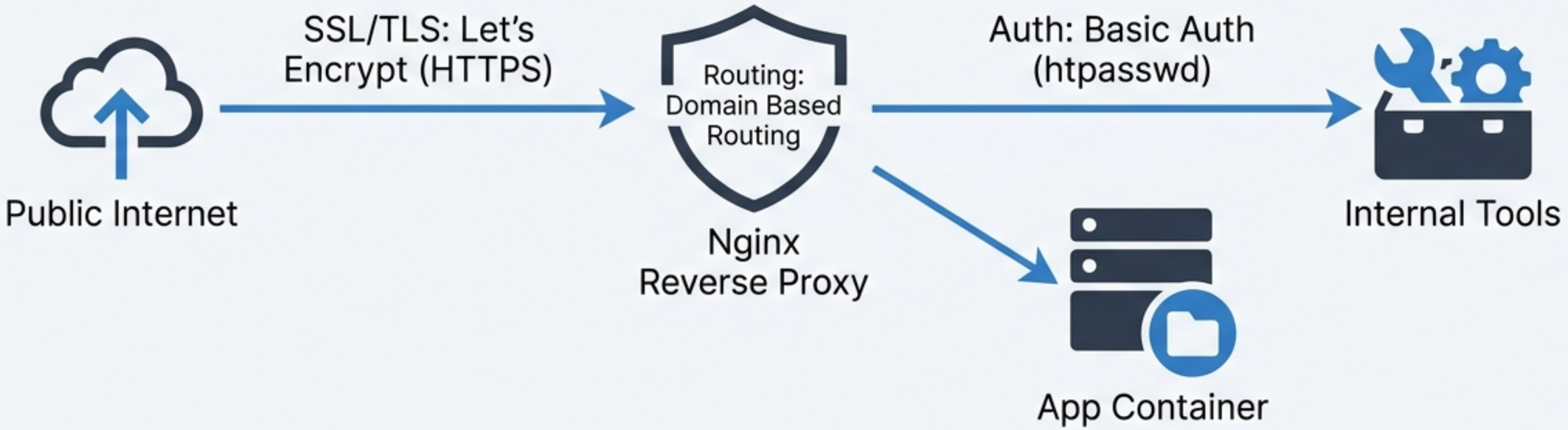
→ **RAM > 90%** →

- **Alert Rules**: Definisi threshold kritis resource.

Real-time Notification: Telegram Bot

- **Alertmanager**: Manajemen status alert (Firing vs Resolved).
- **Integration**: Notifikasi real-time ke Telegram Bot.

Security & Access Management

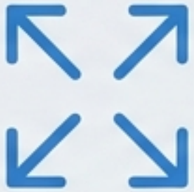


- Reverse Proxy: Nginx routing ke container internal.
- Encryption: Enkripsi traffic HTTPS otomatis.
- Protection: Pembatasan akses ke layanan monitoring.

Ringkasan & Hasil Akhir

Fully Automated

Zero-downtime potential deployment.



Scalable

Infrastruktur AWS siap scale-up.

Observable

Monitoring & Alerting 24/7.

Secure

SSL & Access Management.

Infrastruktur modern yang aman, otomatis, dan terukur.