

UD06 · Prometheus + Grafana

Chuleta rápida de monitorización de infraestructura IaC con Terraform, Prometheus y Grafana.

⌚ 1 · Monitorización en IaC

Observar el estado de los sistemas es esencial en Infraestructura como Código (IaC). Terraform despliega infra, pero **no monitoriza**. Aquí es donde entran Prometheus y Grafana:

- 📊 Rendimiento (CPU, RAM, red...)
- 🌐 Disponibilidad (estado de servicios, contenedores...)
- ⚠️ Alertas tempranas y logs históricos

📈 2 · Prometheus – Recolección de Métricas

- 🤖 Modelo **pull**: Prometheus recoge métricas desde endpoints
- SQLite Base de datos interna optimizada (sin dependencias externas)
- 🌐 Lenguaje de consultas **PromQL**
- ⚙️ Exporters: `node_exporter`, `cAdvisor`, etc.

⚙️ Ejemplo `prometheus.yml`

```
global:  
  scrape_interval: 10s  
  
scrape_configs:  
  - job_name: "node_exporter"  
    static_configs:  
      - targets: ["node_exporter:9100"]  
  
  - job_name: "prometheus"  
    static_configs:  
      - targets: ["prometheus:9090"]
```

📊 3 · Grafana – Visualización

- 🕒 Dashboards exportables en JSON
- 🔗 Compatible con Prometheus, InfluxDB, Loki, etc.
- ⚠️ Alertas visuales o por Slack/webhook
- 👤 Roles y control de acceso

🔊 Ejemplo PromQL para CPU

```
rate(node_cpu_seconds_total{mode!="idle"}[1m])
```

🐳 4 · Terraform + Docker

Terraform despliega infra y contenedores. Así definimos Prometheus + Grafana:

📦 Prometheus

```
resource "docker_container" "prometheus" {  
  image = "prom/prometheus:latest"  
  ports {  
    internal = 9090  
    external = 9090  
  }  
  mounts {  
    type   = "bind"  
    source = abspath("${path.module}/prometheus.yml")  
    target = "/etc/prometheus/prometheus.yml"  
  }  
}
```

🔊 Grafana

```
resource "docker_container" "grafana" {  
  image = "grafana/grafana:latest"  
  ports {  
    internal = 3000  
    external = 3000  
  }  
  depends_on = [docker_container.prometheus]  
  env = [  
    "GF_SECURITY_ADMIN_USER=admin",  
    "GF_SECURITY_ADMIN_PASSWORD=admin"  
  ]  
}
```

```
    ]  
}
```

❖ 5 · Flujo de datos e integración

- ❖ **node_exporter** → métricas expuestas en `:9100/metrics`
- ❖ Prometheus las recoge y almacena
- ❖ Grafana consulta a Prometheus vía HTTP

💡 Se puede automatizar todo en Terraform, incluso dashboards (JSON) con provisión automática.

🔗 Dashboard JSON básico

```
{  
  "title": "WordPress Basic Metrics",  
  "panels": [  
    {  
      "type": "graph",  
      "title": "CPU Usage",  
      "targets": [  
        {  
          "expr": "rate(node_cpu_seconds_total{mode!='idle'})[1m]",  
          "legendFormat": "CPU"  
        }  
      ]  
    }  
  ]  
}
```

💡 6 · Buenas prácticas

- 📁 Versiona dashboards JSON en Git
- 🔒 Usa variables para contraseñas (no en `main.tf`)
- 🧩 Separa módulos Terraform: `modules/monitoring`
- /docker/ Usa volúmenes Docker para persistencia de métricas

☞ Cheat Sheet UD06 – Prometheus + Grafana · Full Edition