Per-Folder Usage Dashboard: Grafana + Prometheus + Node Exporter

This manual sets up a clean pipeline for visualizing per-folder storage usage for team directories like /data/proj1…/proj4. We leverage Node Exporter's Textfile Collector to ingest a custom metric (folder bytes). Prometheus scrapes it; Grafana graphs it.
Note: This approach is lightweight and safe. You do not need root on Grafana. Only the metric emitter script and node_exporter need sudo/systemd changes.

Architecture (1-minute overview)

1) A small Bash script measures each folder's size and writes Prometheus metrics into a text file.
2) node_exporter (with --collector.textfile.directory) reads that file and exposes the gauges.
3) Prometheus scrapes node_exporter.
4) Grafana connects to Prometheus and renders panels + alerts.

1) Install node_exporter (with textfile collector)

```
# Download and install node_exporter (Debian/Ubuntu example)
NODE_VER="1.8.2"
cd /tmp
curl -LO https://github.com/prometheus/node_exporter/releases/download/v${NODE_VER}/node_expo
tar xzf node_exporter-${NODE_VER}.linux-amd64.tar.gz
sudo mv node_exporter-${NODE_VER}.linux-amd64/node_exporter /usr/local/bin/

# Textfile directory for custom metrics
sudo useradd --no-create-home --shell /usr/sbin/nologin nodeexp || true
sudo install -d -o nodeexp -g nodeexp -m 755 /var/lib/node_exporter/textfile

# Systemd unit
cat << 'EOF' | sudo tee /etc/systemd/system/node_exporter.service
[Unit]
Description=Node Exporter
After=network.target

[Service]
User=nodeexp
Group=nodeexp
ExecStart=/usr/local/bin/node_exporter \
  --collector.textfile \
  --collector.textfile.directory=/var/lib/node_exporter/textfile
Restart=always

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable --now node_exporter
```

2) Emit per-folder metrics (textfile collector script)

Save as /usr/local/bin/folder_metrics.sh and run via systemd timer or cron. It writes gauges like storage_folder_bytes{folder="/data/proj1"} 12345

```
#!/usr/bin/env bash
# /usr/local/bin/folder_metrics.sh
set -euo pipefail

OUT="/var/lib/node_exporter/textfile/folder_usage.prom.$$"
```

```
DEST="/var/lib/node_exporter/textfile/folder_usage.prom"

FOLDERS=(/data/proj1 /data/proj2 /data/proj3 /data/proj4)
TS=$(date +%s)

# Compute sizes in bytes (fast and scriptable)
for d in "${FOLDERS[@]}"; do
  if [[ -d "$d" ]]; then
    BYTES=$(du -sb "$d" 2>/dev/null | awk '{print $1}')
    echo "storage_folder_bytes{folder=\"${d}\"} ${BYTES}" >> "$OUT"
    echo "storage_folder_timestamp_seconds{folder=\"${d}\"} ${TS}" >> "$OUT"
  fi
done

# Atomic move into place so node_exporter never reads a partial file
sudo chown nodeexp:nodeexp "$OUT" || true
mv "$OUT" "$DEST"
```

Make it executable:

```
sudo chmod +x /usr/local/bin/folder_metrics.sh
```

3) Schedule the metrics script (systemd timer)

```
# Service
cat << 'EOF' | sudo tee /etc/systemd/system/folder-metrics.service
[Unit]
Description=Emit per-folder usage metrics

[Service]
Type=oneshot
ExecStart=/usr/local/bin/folder_metrics.sh
EOF

# Timer (every 5 minutes)
cat << 'EOF' | sudo tee /etc/systemd/system/folder-metrics.timer
[Unit]
Description=Run folder-metrics every 5 minutes

[Timer]
OnBootSec=1min
OnUnitActiveSec=5min
AccuracySec=30s
Unit=folder-metrics.service

[Install]
WantedBy=timers.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable --now folder-metrics.timer
```

4) Install Prometheus and add a scrape job

```
# Debian/Ubuntu quick install
sudo apt-get update && sudo apt-get install -y prometheus

# Add a scrape job for node_exporter (edit /etc/prometheus/prometheus.yml)
# Example target is localhost:9100 (default node_exporter port)
# Add under 'scrape_configs:'

  - job_name: 'node_exporter'
    static_configs:
```

```
      - targets: ['localhost:9100']

# Then reload/restart:
sudo systemctl restart prometheus
```

5) Install Grafana and add Prometheus datasource

```
# Debian/Ubuntu quick install (OSS)
sudo apt-get install -y adduser libfontconfig1 musl
curl -L https://dl.grafana.com/oss/release/grafana_11.0.0_amd64.deb -o /tmp/grafana.deb
sudo dpkg -i /tmp/grafana.deb
sudo systemctl enable --now grafana-server

# In Grafana UI:
#  - Login (default: admin/admin)
#  - Add Data Source -> Prometheus -> URL http://localhost:9090 -> Save & Test
```

6) Build the dashboard (metrics & queries)

Create a new dashboard with the following panels and PromQL queries:

```
# Panel: Current Usage (per folder)
Query: storage_folder_bytes

# Panel: Usage Trend Over Time
Query: storage_folder_bytes

# Table: Top N Folders by Usage (if you add more)
Query: topk(10, storage_folder_bytes)

# SingleStat: Total Used by All Teams
Query: sum(storage_folder_bytes)

# (If you configured per-team hard caps, store them as constants or as recording rules to co
# Example % used with 25 GB per folder:
# Query: 100 * storage_folder_bytes / (25 * 1024 * 1024 * 1024)
```

7) Alerts (optional)

Prometheus alerting rule example (90% threshold). Save under /etc/prometheus/rules/folder_usage.yml and include it from prometheus.yml.

```
groups:
- name: folder-usage
  rules:
  - alert: FolderUsageHigh
    expr: storage_folder_bytes / (25 * 1024 * 1024 * 1024) > 0.90
    for: 15m
    labels:
      severity: warning
    annotations:
      summary: "Folder usage above 90%: {{ $labels.folder }}"
      description: "{{ $labels.folder }} is at {{ $value | humanizePercentage }} of its quot
```

8) Security & hardening tips

• Restrict Grafana/Prometheus to your VPN or internal subnets.
• Give node_exporter a dedicated non-privileged user (already done).
• The metric script only writes to the textfile directory; keep it root-owned and not world-writable.
• Consider adding a service monitor (systemd watchdog or Monit) to ensure exporters and timers stay healthy.

Appendix: Using cron instead of systemd timer

```
# Run every 5 minutes (root)
sudo crontab -e
*/5 * * * * /usr/local/bin/folder_metrics.sh
```

Appendix: Adding more folders

Just edit FOLDERS=(...) in both the email script and the metrics script. You can template these from Ansible if you manage many servers.