

# Prometheus Cheatsheet

## Connection

**Access Prometheus Web UI**  
`http://localhost:9090`

**curl API endpoint**  
`curl http://localhost:9090/api/v1/query?query=up`

**Configure Prometheus**  
Edit `prometheus.yml` file

**Start Prometheus**  
`prometheus --config.file=prometheus.yml`

## Querying

**Basic query**  
`http_requests_total`

**Filter by label**  
`http_requests_total{status="200"}`

**Range query**  
`http_requests_total[5m]`

**Rate**  
`rate(http_requests_total[5m])`

**Sum by**  
`sum by(instance) (http_requests_total)`

**Increase**  
`increase(http_requests_total[1h])`

**Aggregation functions**  
`sum()`, `avg()`, `min()`, `max()`, `count()`

## Alert Management

**View alerts**  
`http://localhost:9090/alerts`

**List alertmanager targets**  
`http://localhost:9090/targets`

**Define alert rule**  
In `prometheus.yml` or rules file

## Management

**List targets**  
`http://localhost:9090/targets`

**View configuration**  
`http://localhost:9090/config`

**View service discovery**  
`http://localhost:9090/service-discovery`

**Runtime information**  
`http://localhost:9090/status`

## PromQL (Advanced)

**Quantile**  
`histogram_quantile(0.95, rate(http_request_duration_seconds_bucket[5m]))`

**Predict linear trend**  
`predict_linear(node_filesystem_free_bytes[1h], 4 * 3600)`

**Counter reset handling**  
`resets(http_requests_total[5m])`

**Conversion to per-second**  
`irate(http_requests_total[5m])`

**Delta for gauges**  
`delta(temperature_celsius[1h])`

## CLI Options

**Specify config file**  
`--config.file=prometheus.yml`

**Set storage path**  
`--storage.tsdb.path=data/`

**Retention time**  
`--storage.tsdb.retention.time=15d`

**HTTP listen address**  
`--web.listen-address=0.0.0.0:9090`

**External URL**  
`--web.external-url=http://prometheus.example.org`