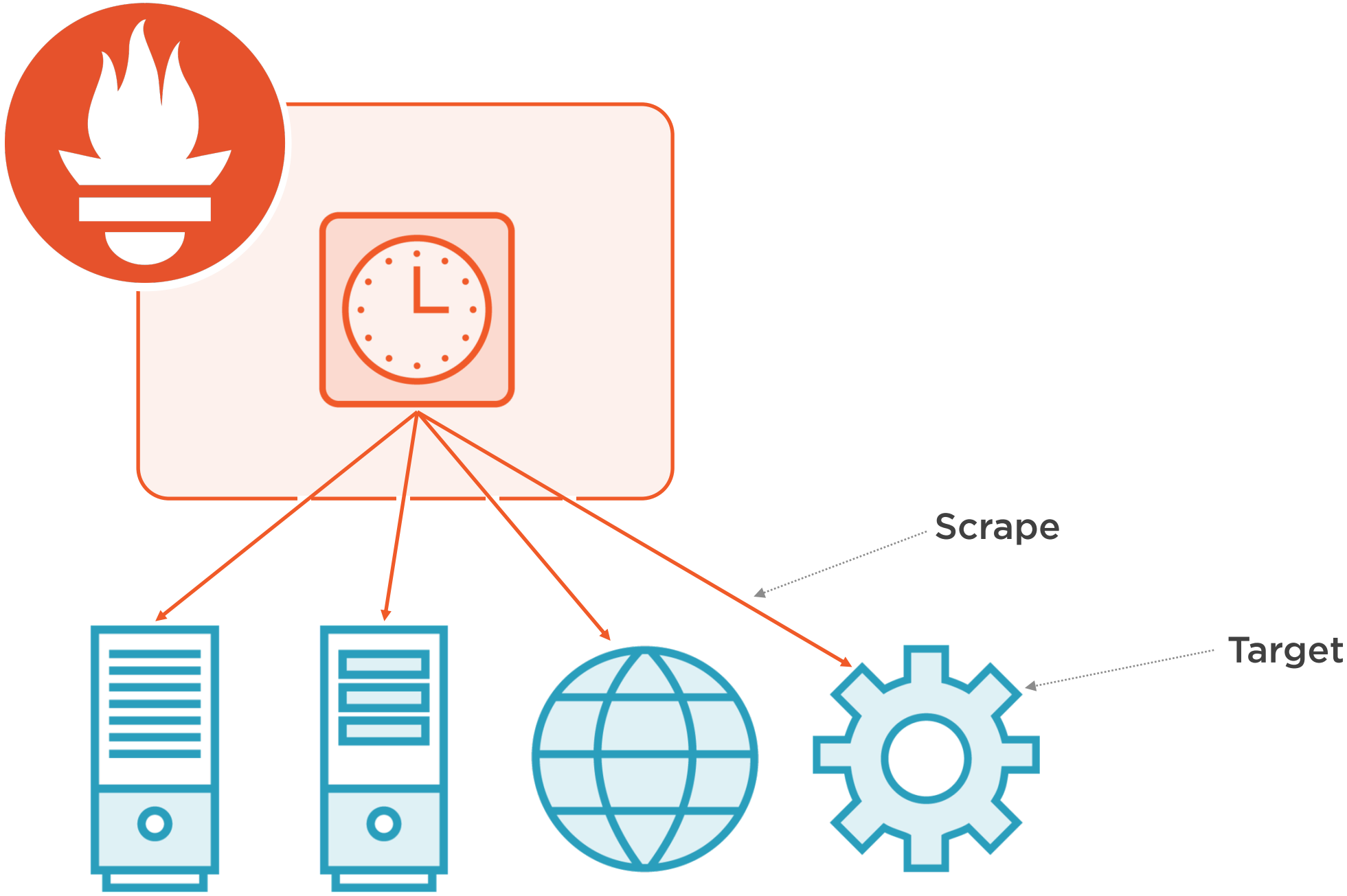
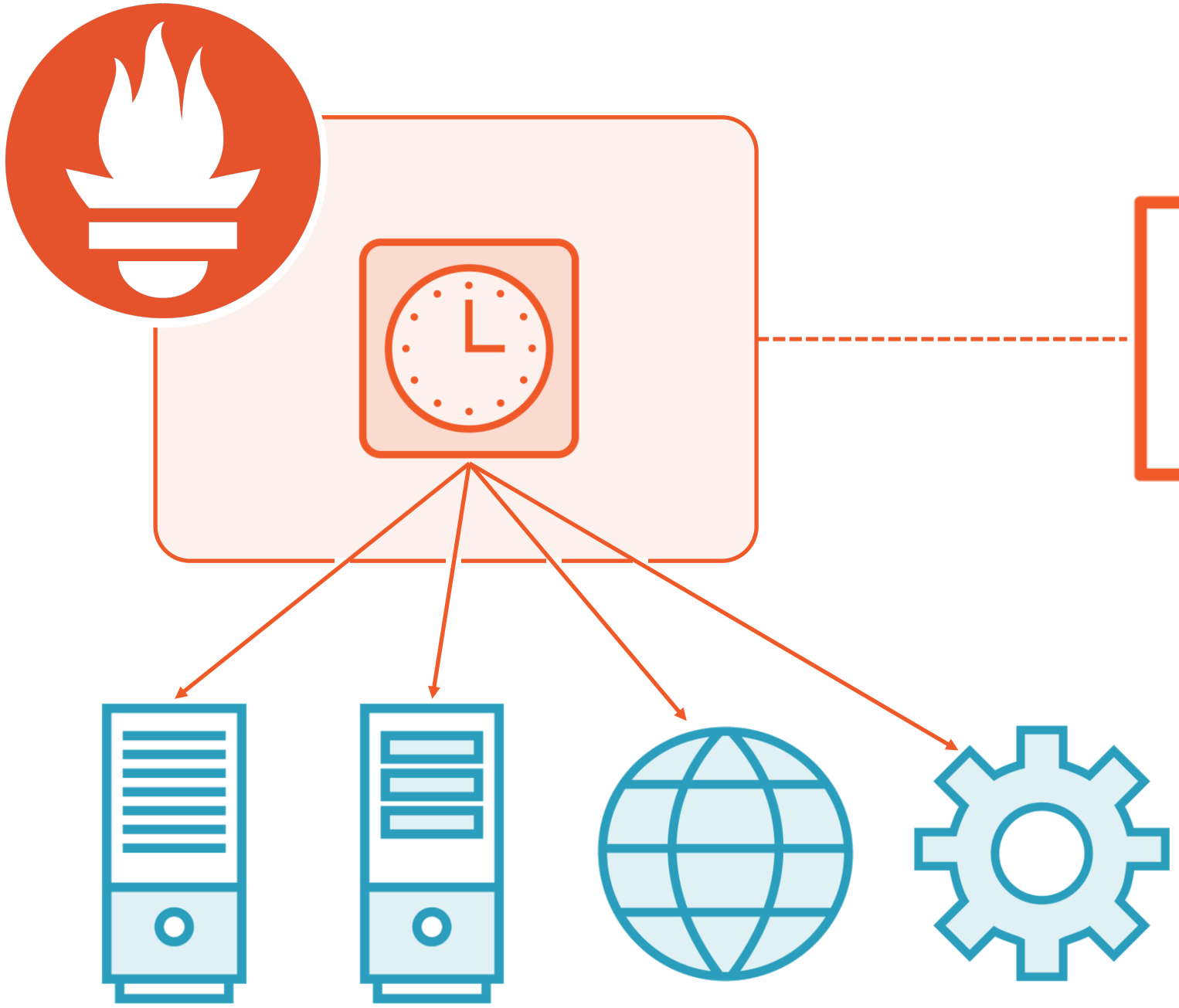
Running and Configuring Prometheus

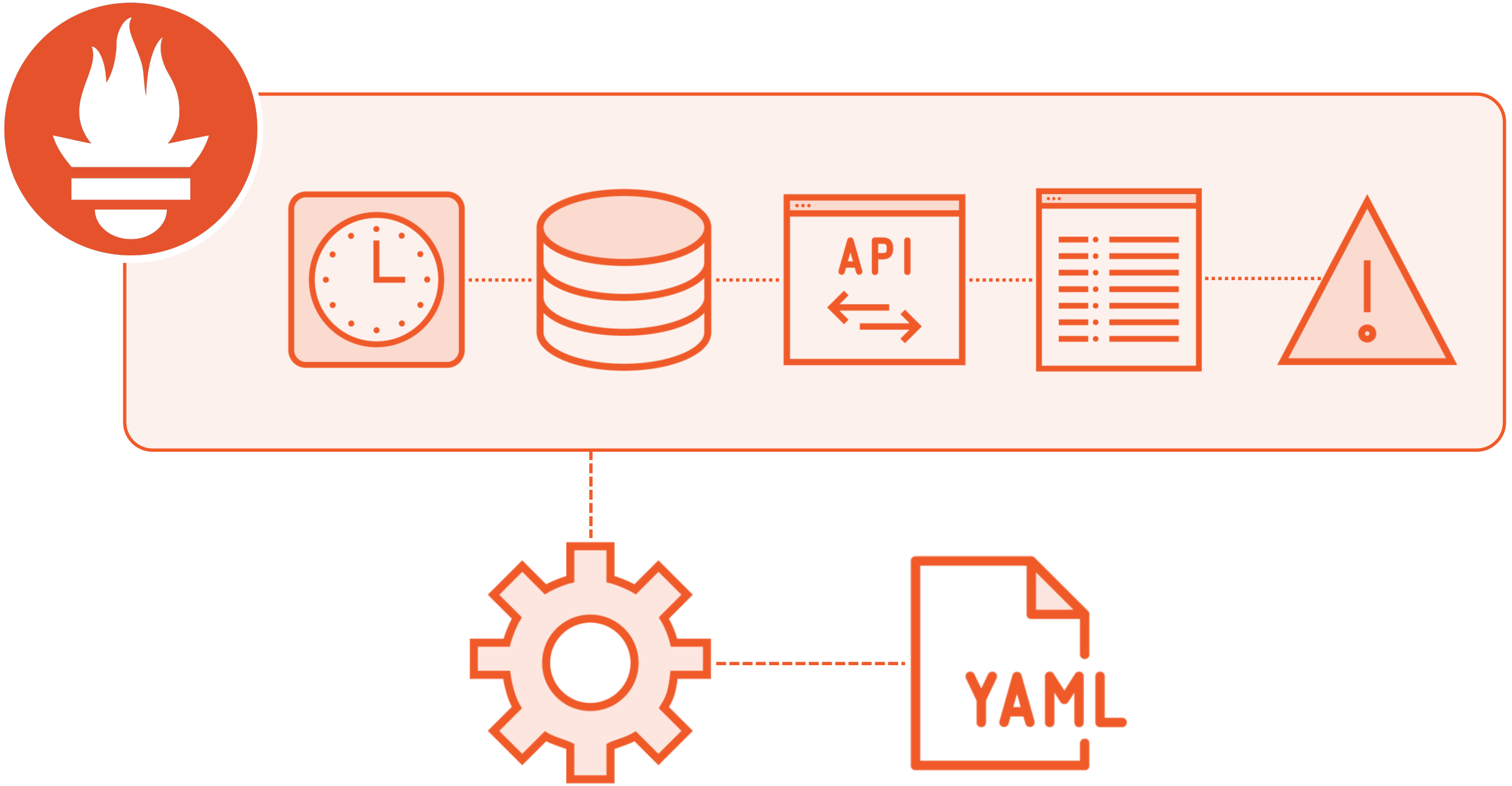
Elton Stoneman
CONSULTANT & TRAINER

@EltonStoneman | blog.sixeyed.com


Scrape
Target



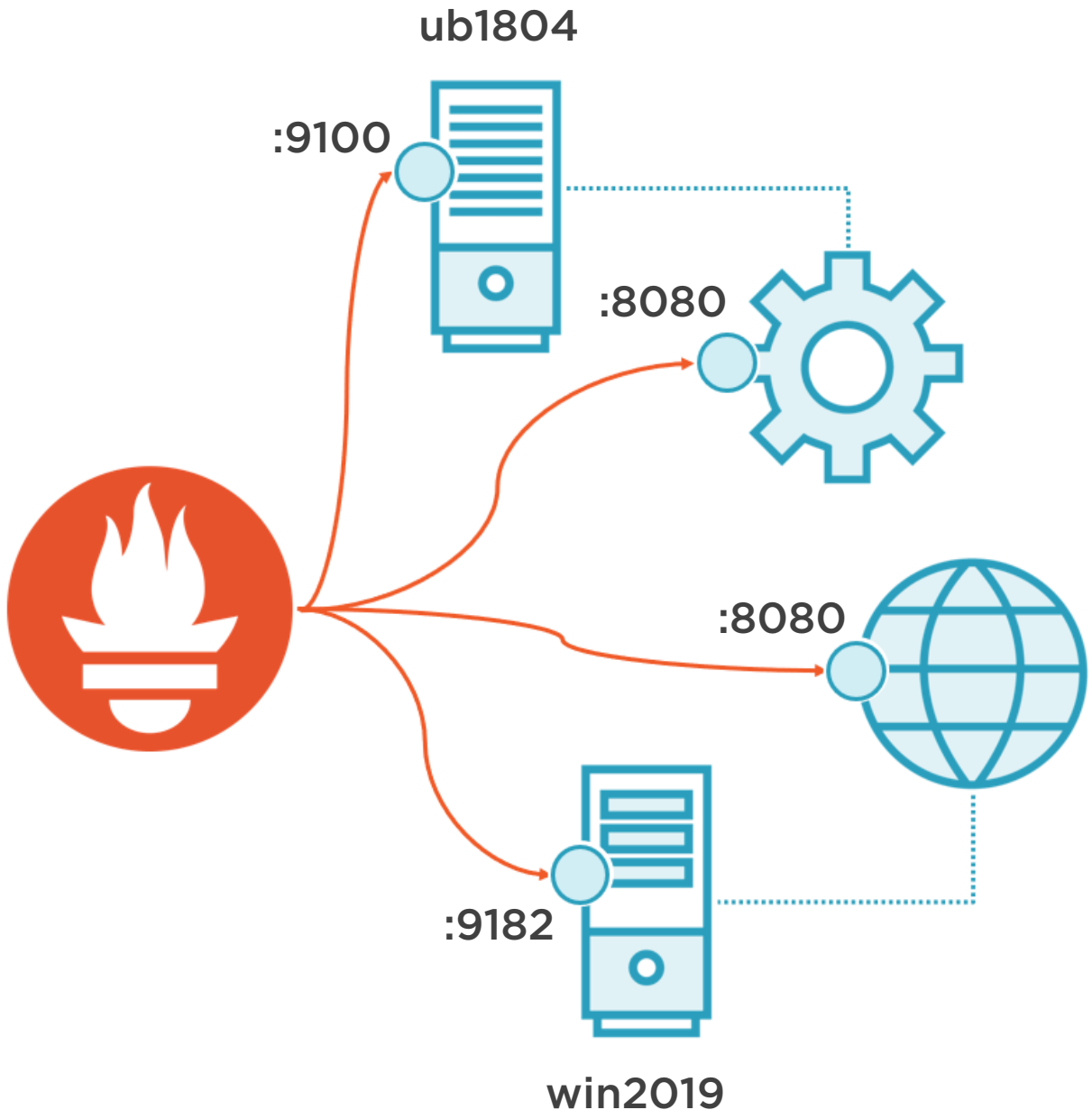
- Targets
- Scrape interval
- Metric updates


API
YAML

prometheus.yml

```
global:
  scrape_interval: 30s

scrape_configs:
  - job_name: 'linux'
    static_configs:
      - targets: ['ps-prom-ub1804:9100']
```

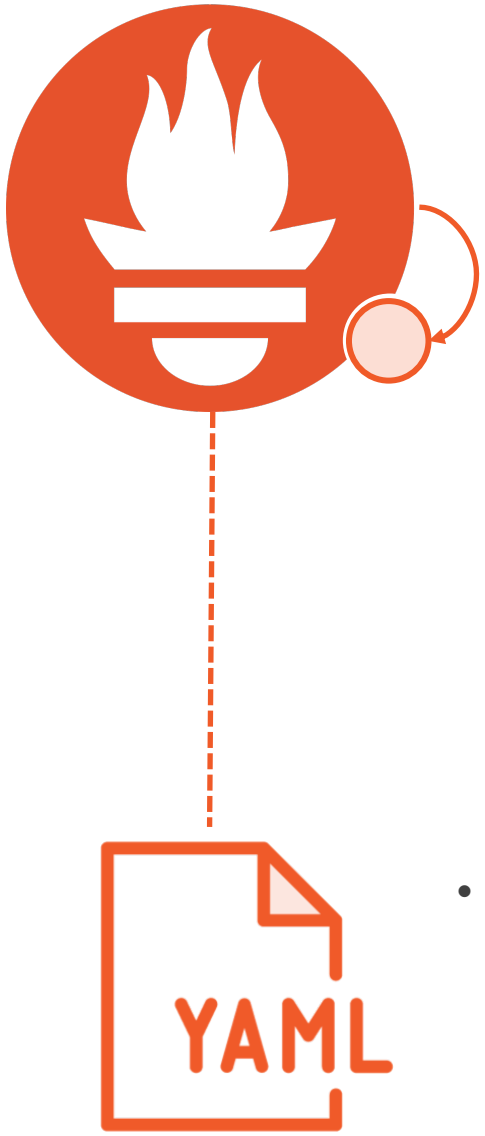
```
scrape_configs:

  - job_name: 'linux'
    static_configs:
      - targets: ['ub1804:9100']

  - job_name: 'batch'
    static_configs:
      - targets: ['ub1804:8080']

  - job_name: 'windows'
    static_configs:
      - targets: ['win2019:9182']

  - job_name: 'web'
    static_configs:
      - targets: ['win2019:8080']
```



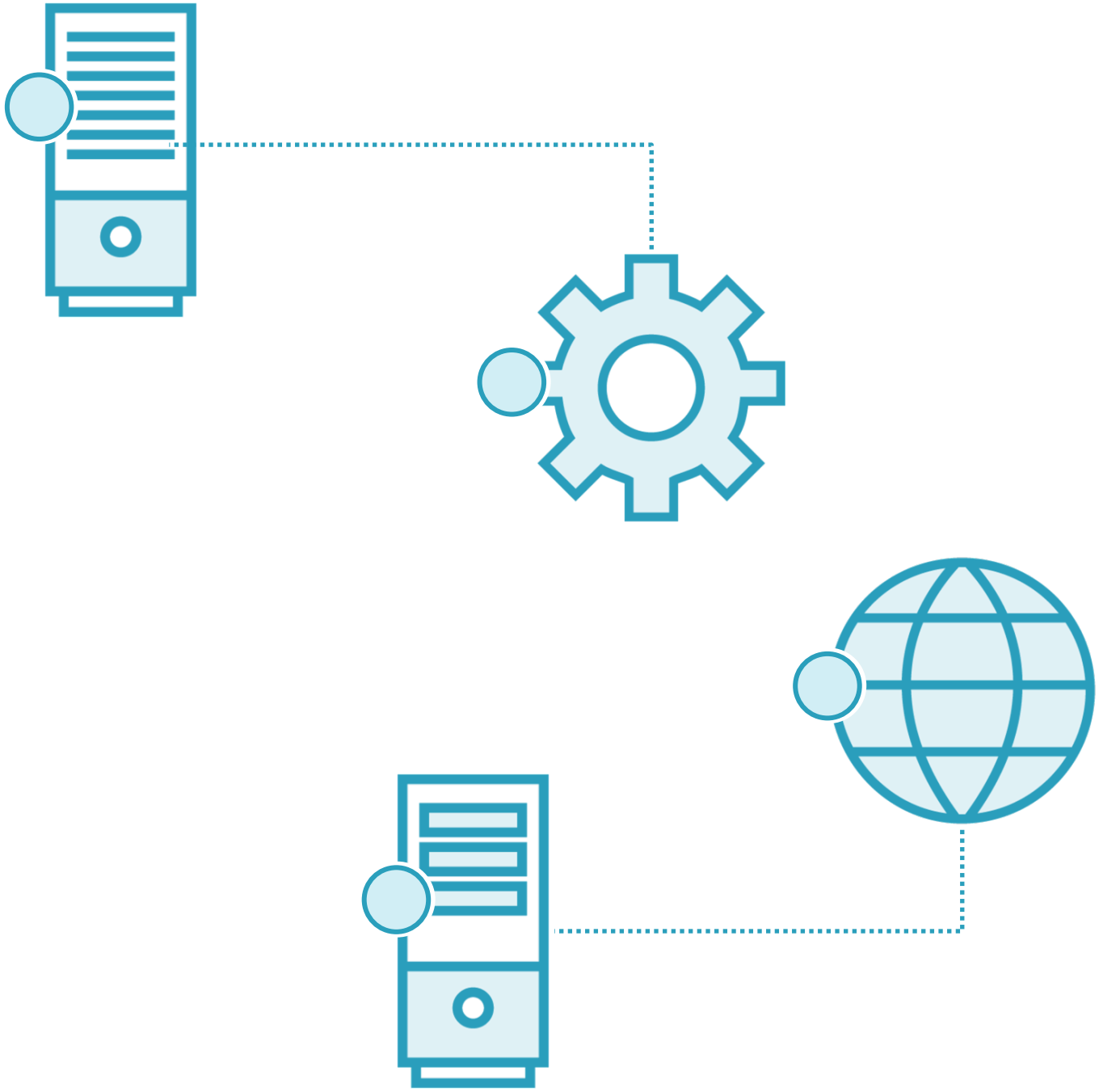
Demo

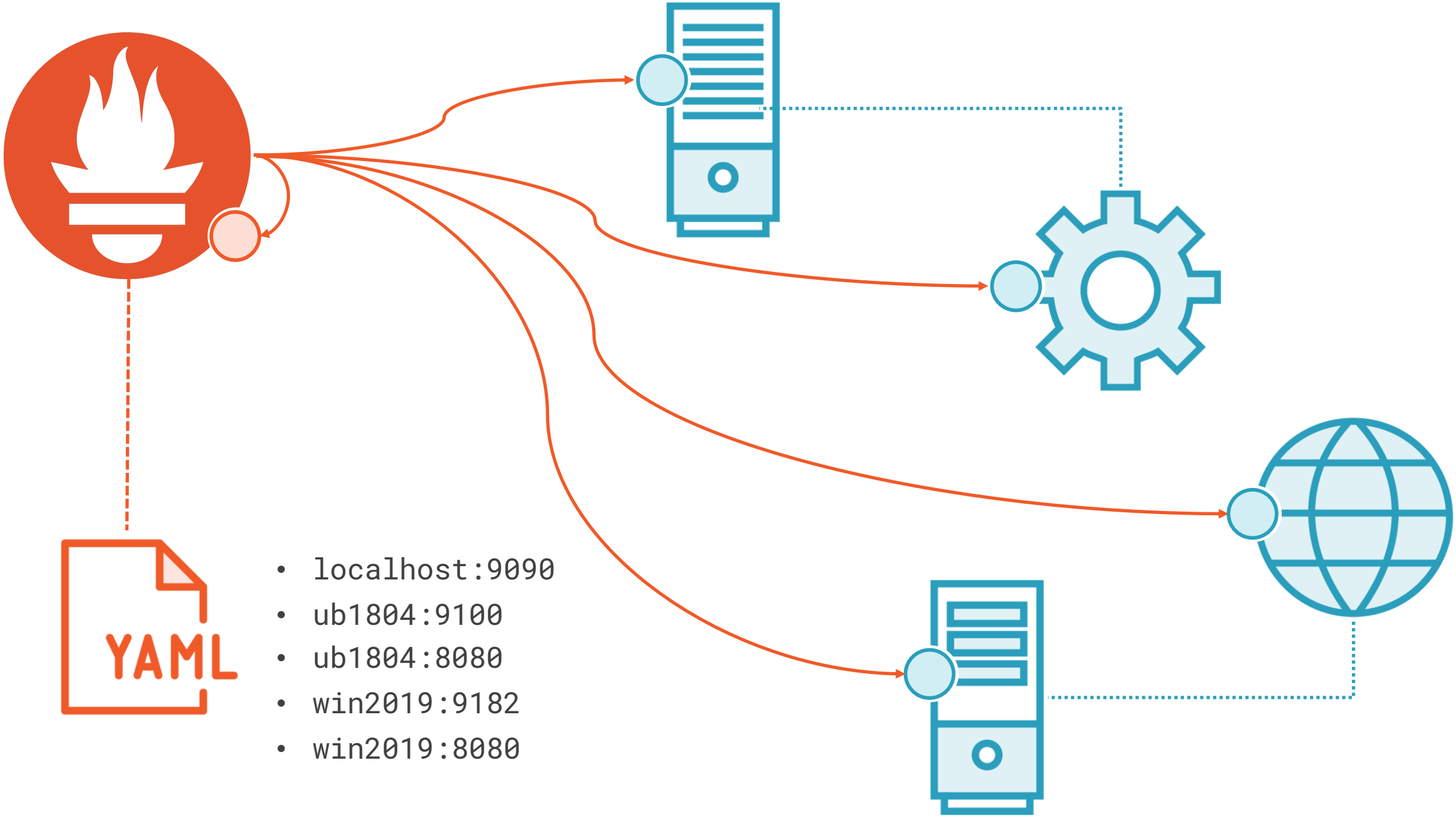
Running Prometheus

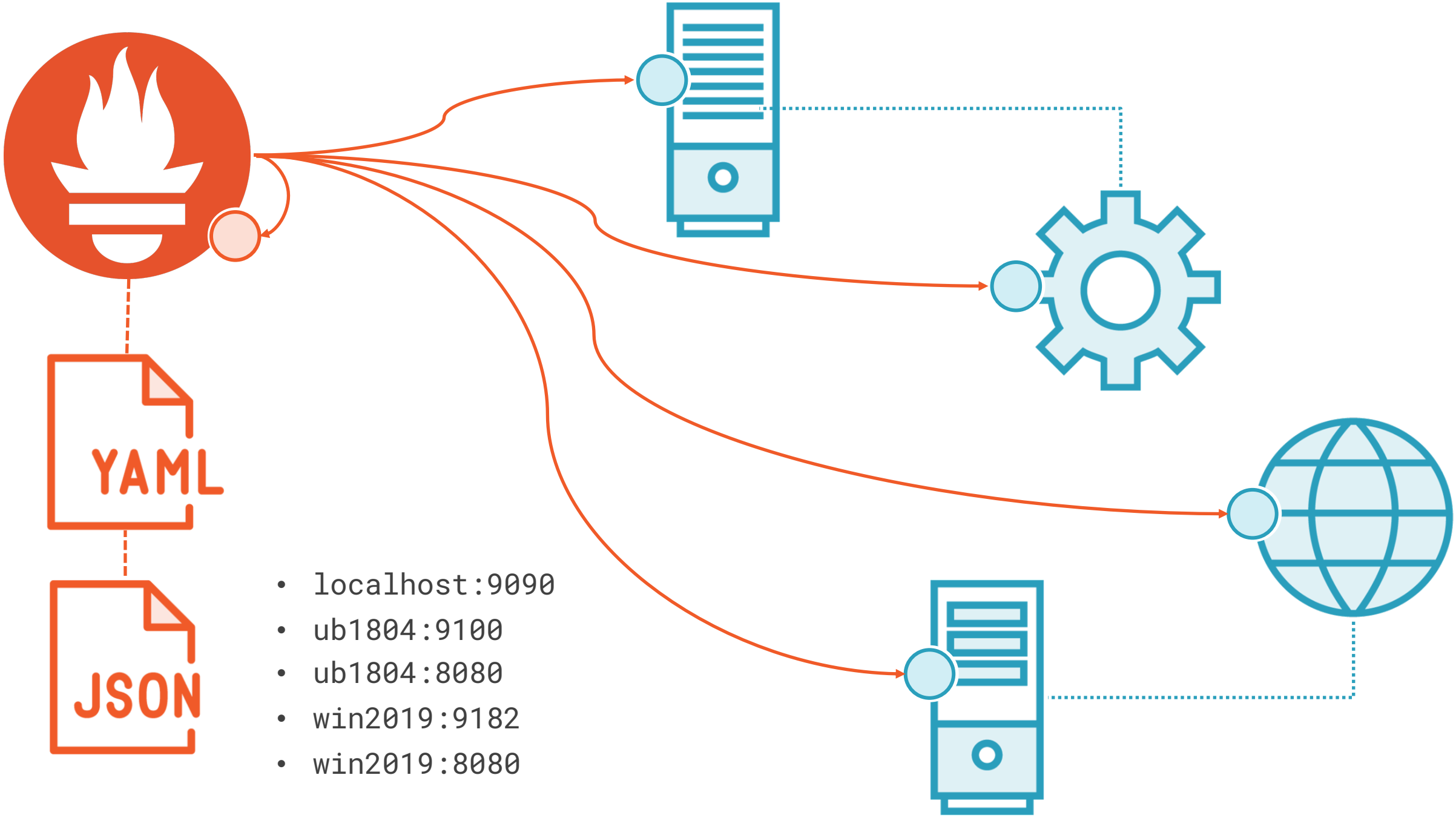
- Download options
- Running the server
- Exploring the UI



- localhost:9090




YAML
localhost:9090
ub1804:9100
ub1804:8080
win2019:9182
win2019:8080

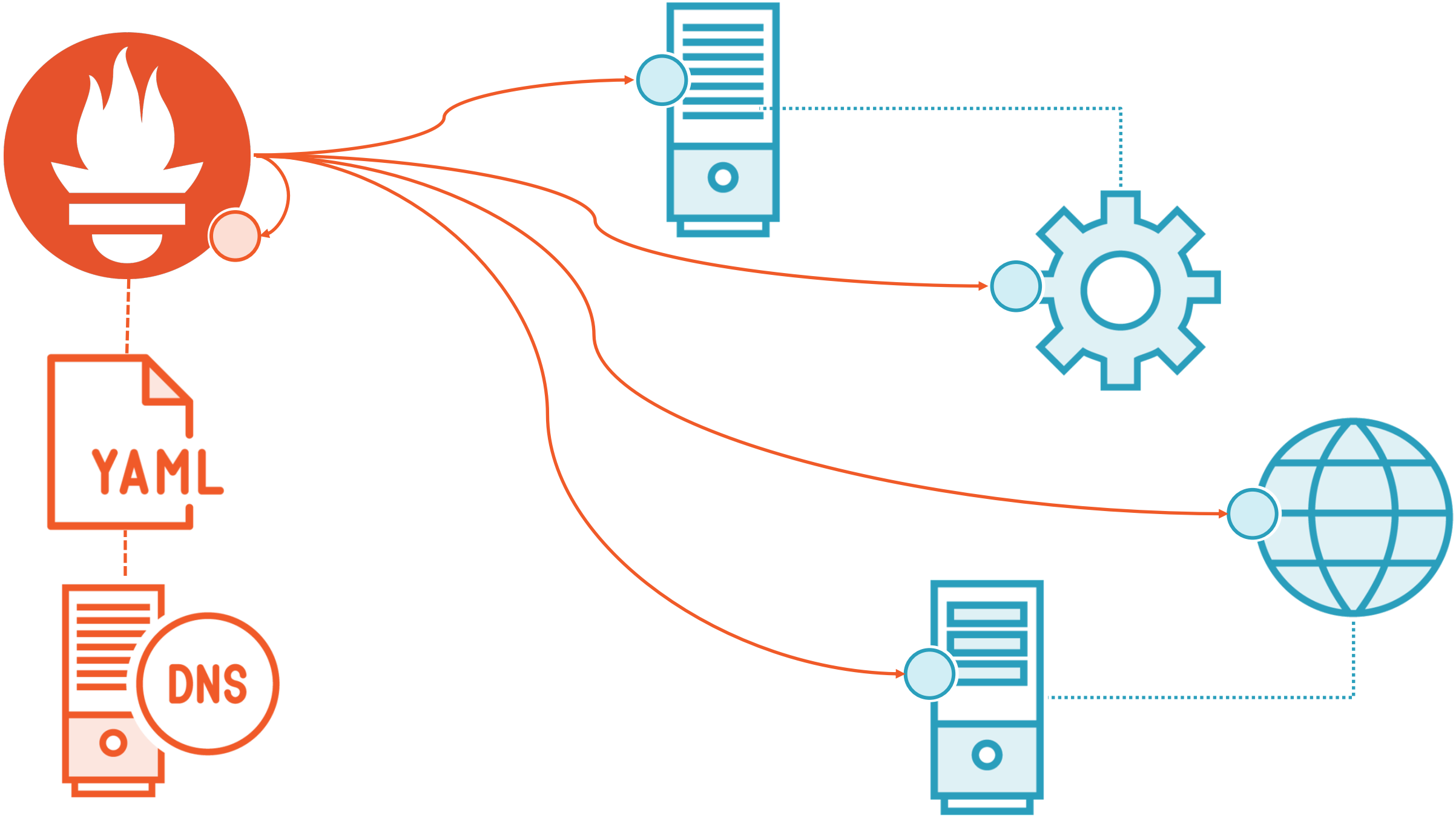

YAML
JSON
• localhost:9090
• ub1804:9100
• ub1804:8080
• win2019:9182
• win2019:8080

file-sd.yml

```yaml
global:
  scrape_interval:     15s

scrape_configs:
  - job_name: prometheus
    static_configs:
    - targets: [localhost:9090]

  - job_name: web-file
    file_sd_configs:
      - files:
          - 'web.json'
```

web.json

```json
[
    {
      "targets": [ "win2019:9182" ]
    },
    {
      "targets": [ "win2019:8080" ]
    }
]
```


YAML
DNS

dns-sd.yml

```yaml
global:
  scrape_interval:     15s

scrape_configs:
  - job_name: web-dns
    dns_sd_configs:
      - names:
        - metrics.web
        type: SRV
```

DNS SRV record

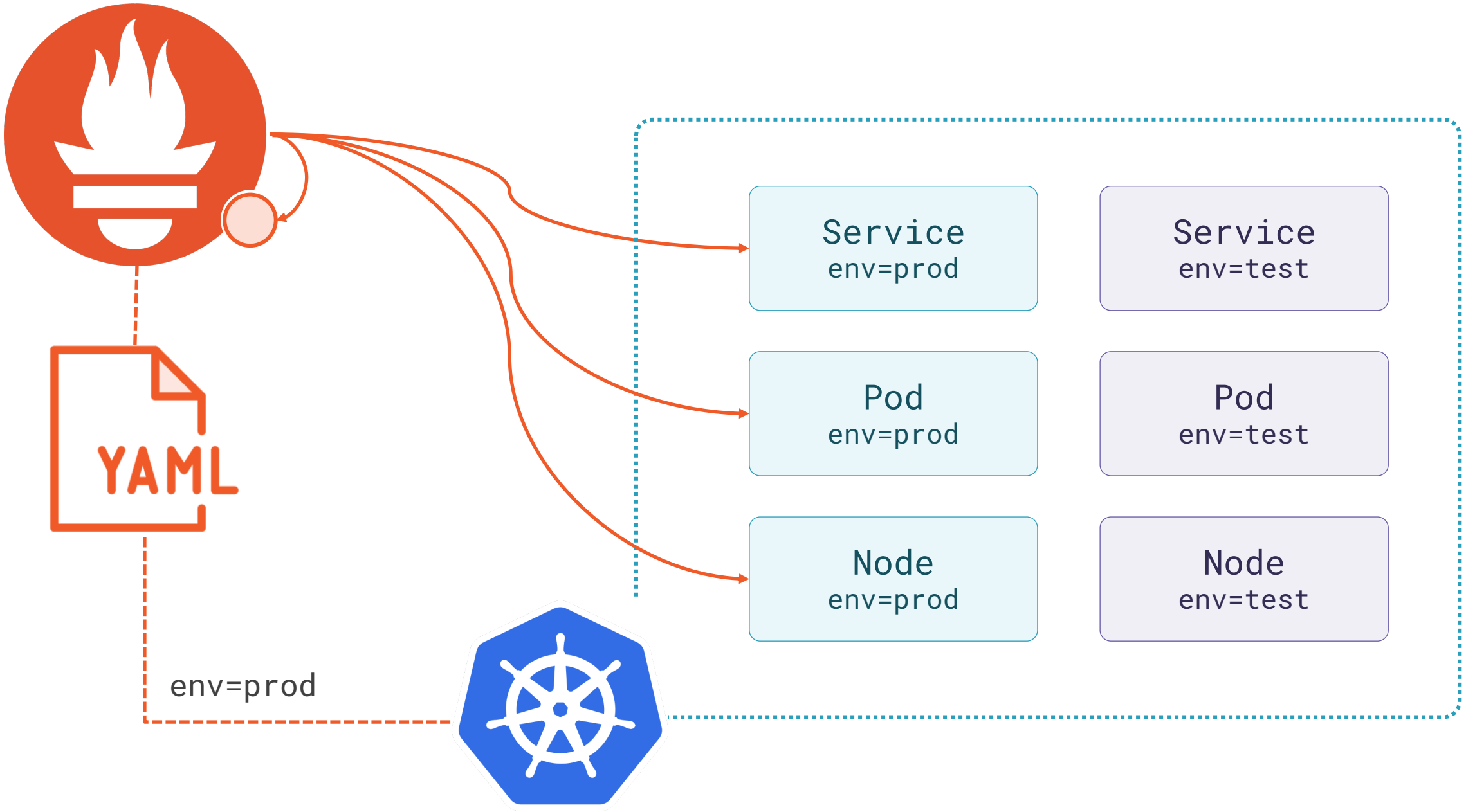
```
# name         TTL   port   target
metrics.web    300   9132   box1.web.tld
metrics.web    300   9132   box2.web.tld
metrics.web    300   9132   box3.web.tld
```

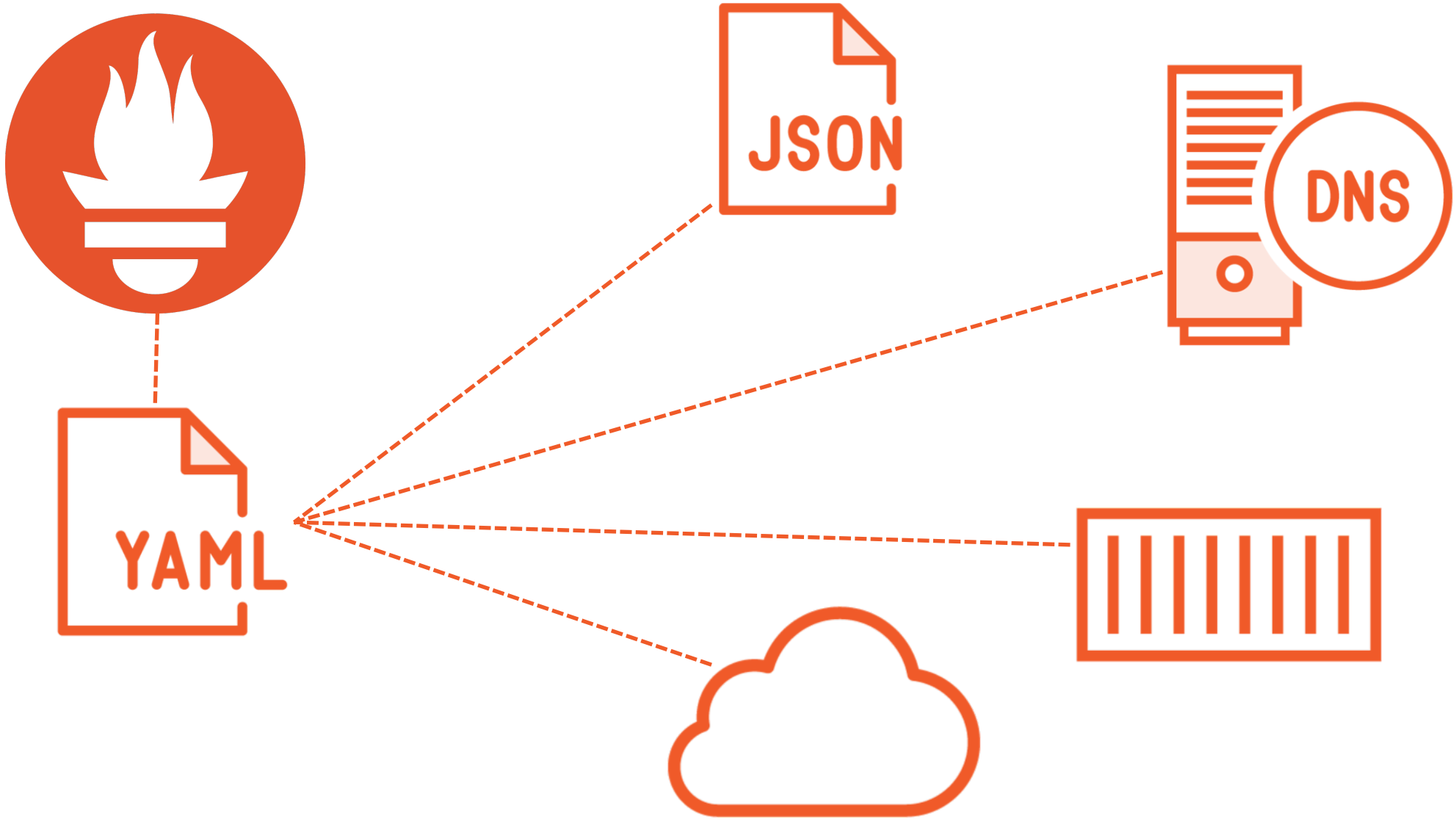
dns-sd.yml

```yaml
global:
  scrape_interval:     15s

scrape_configs:
  - job_name: web-dns
    dns_sd_configs:
      - names:
        - metrics2.web
        type: A
        port: 8080
```

DNS A record

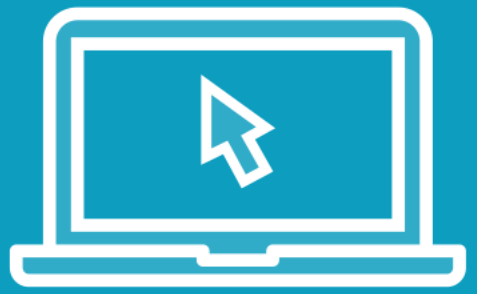
```
# name          address
metrics2.web    172.17.143.229
metrics2.web    172.17.143.236
metrics2.web    172.17.143.242
```


YAML
env=prod
Service
env=prod
Service
env=test
Pod
env=prod
Pod
env=test
Node
env=prod
Node
env=test



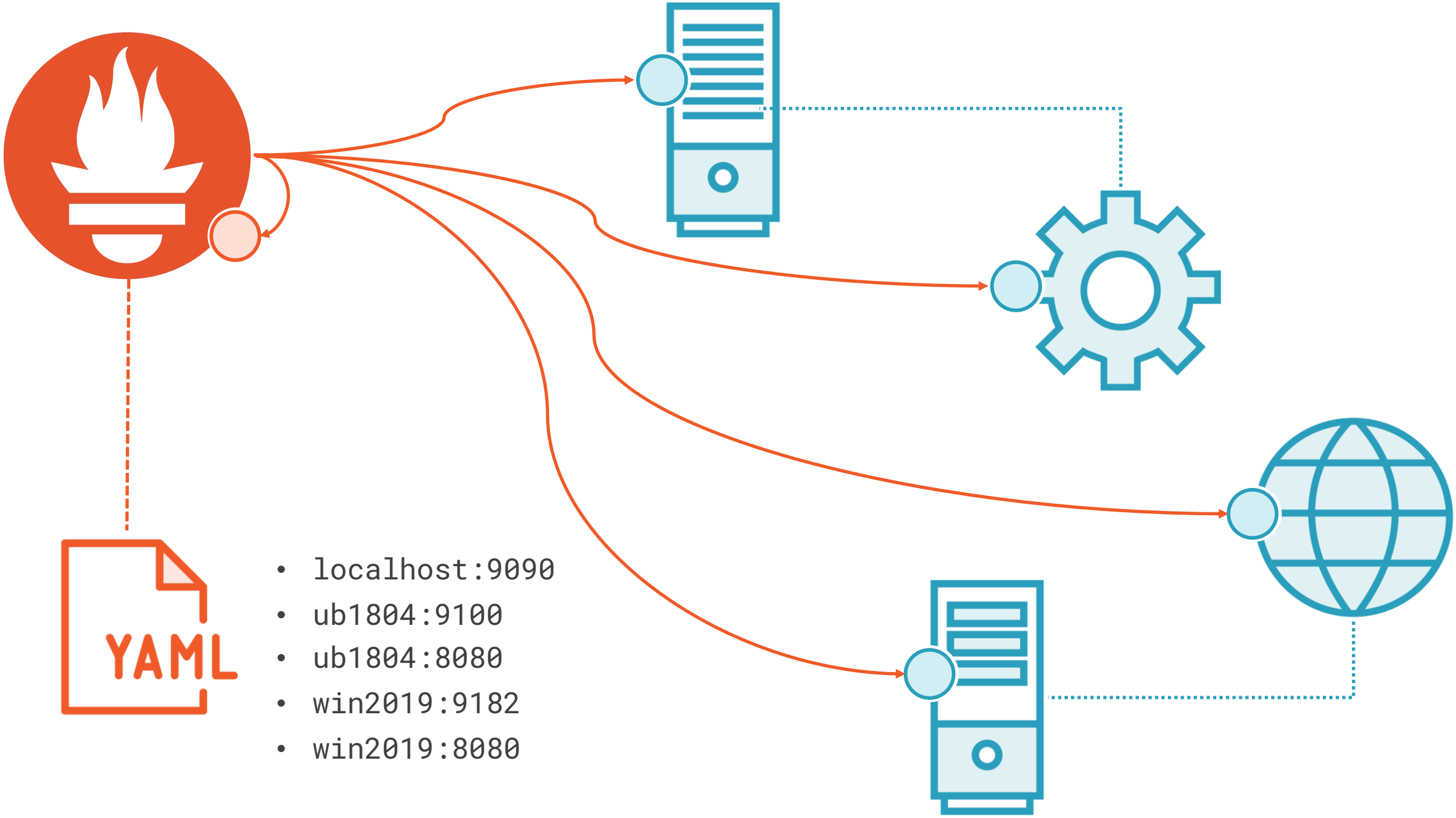
https://is.gd/ovutig

Demo

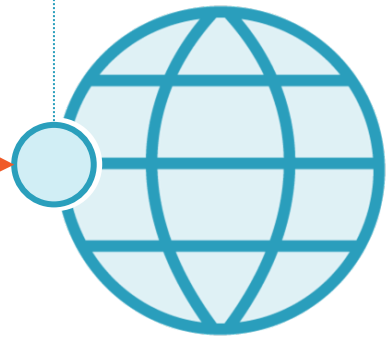


Configuring Prometheus

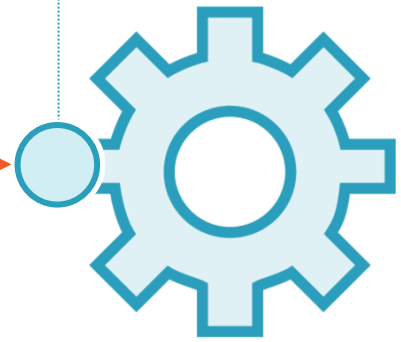
- Static target configuration
- Exploring scraped metrics
- Live configuration reloads

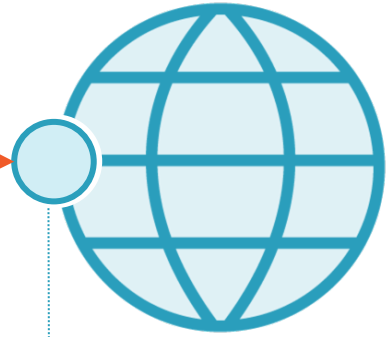

YAML
localhost:9090
ub1804:9100
ub1804:8080
win2019:9182
win2019:8080

```
http_request_duration_seconds_count
{
  code="200",
  method="GET"
}
```



```
http_request_duration_seconds_count
{
  code="200",
  method="GET",
  job="web",
  instance="win2019"
}
```



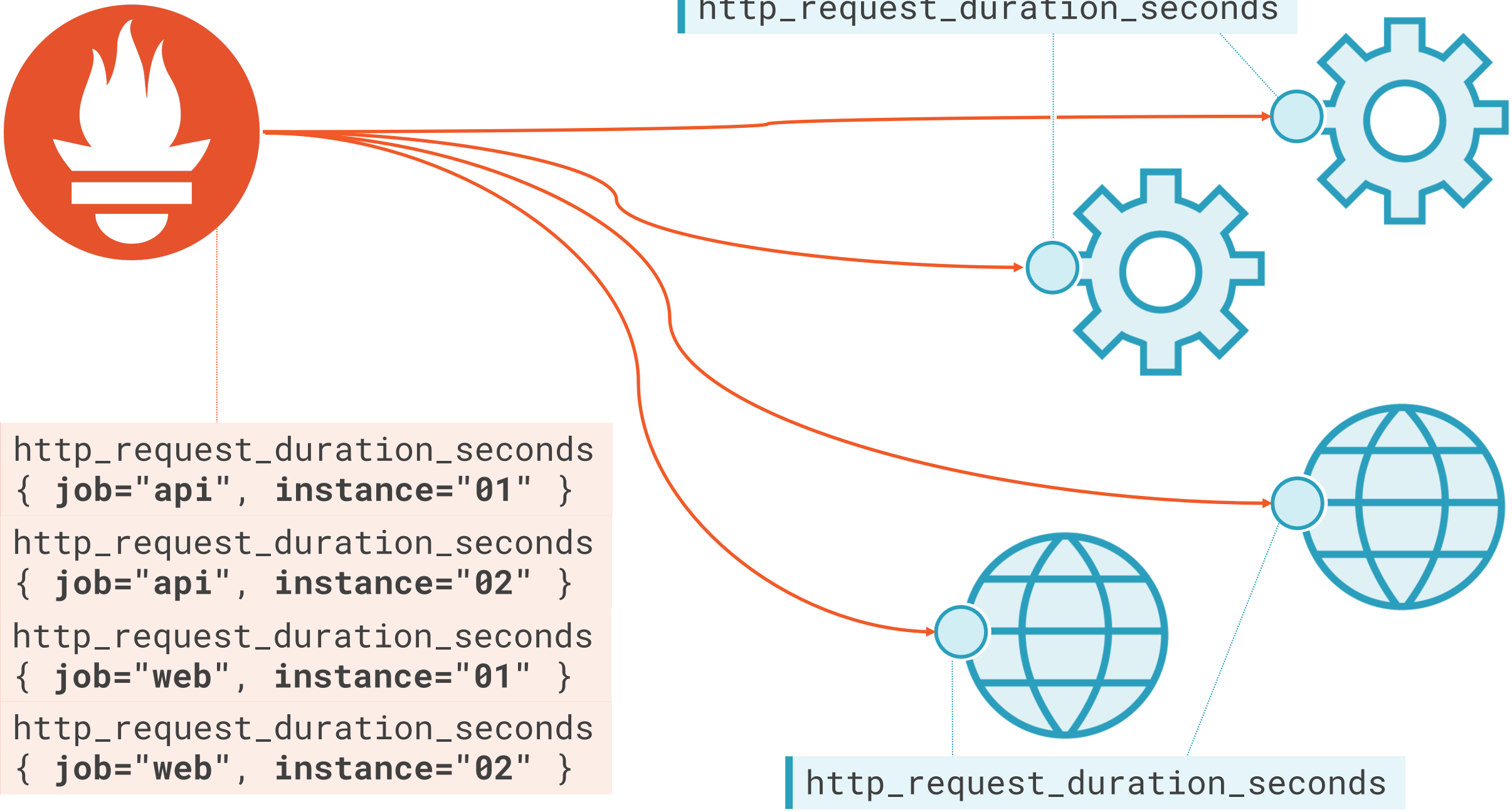


```
http_request_duration_seconds
{ job="api" }
```

```
http_request_duration_seconds
{ job="web" }
```

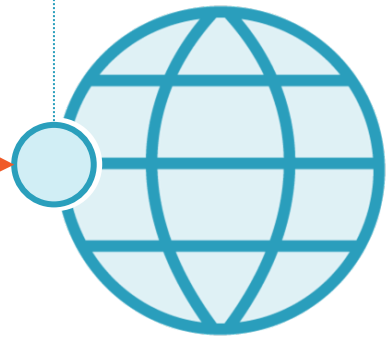
```
http_request_duration_seconds
```

```
http_request_duration_seconds
```

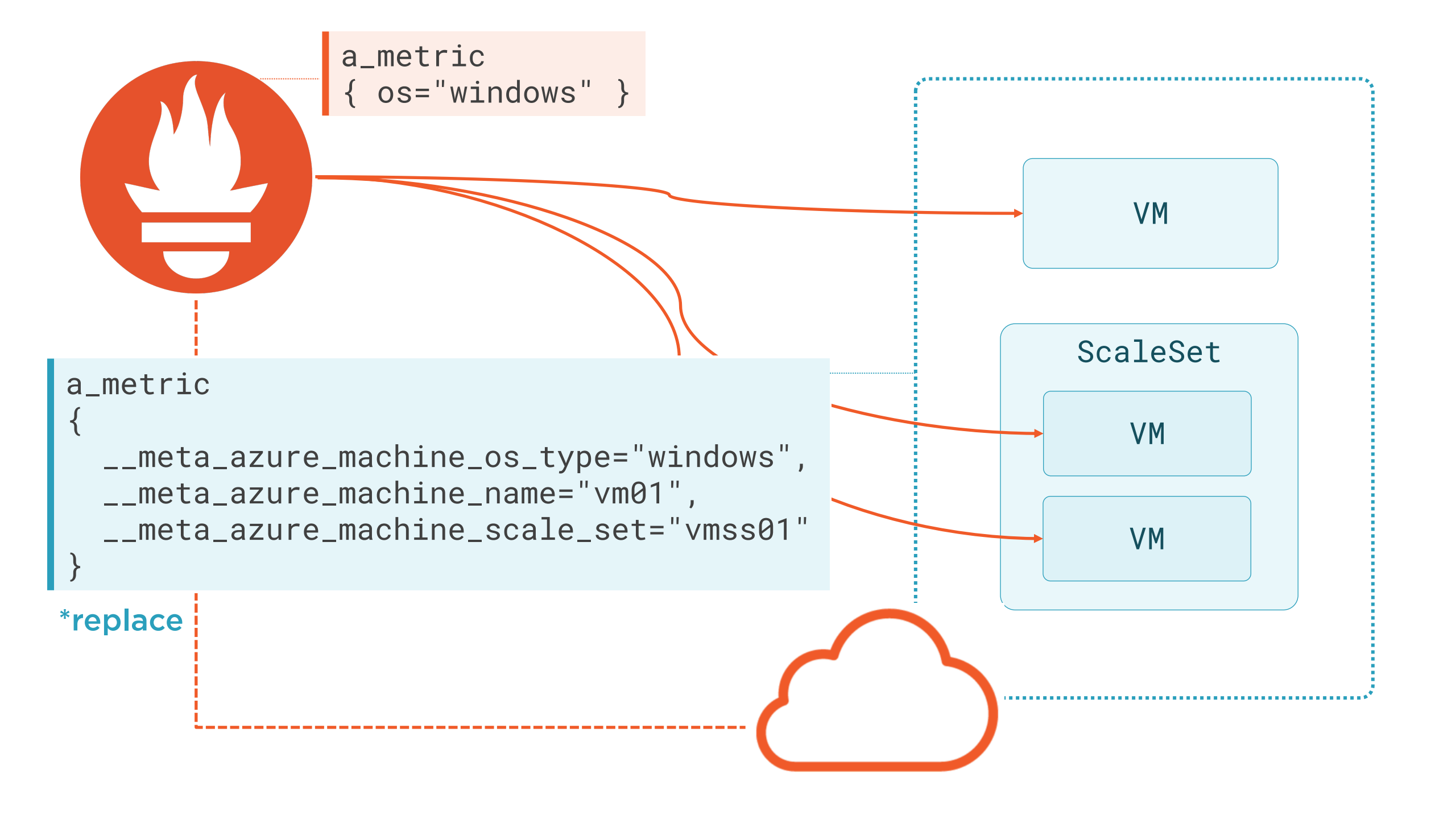

http_request_duration_seconds
http_request_duration_seconds
{ job="api", instance="01" }
http_request_duration_seconds
{ job="api", instance="02" }
http_request_duration_seconds
{ job="web", instance="01" }
http_request_duration_seconds
{ job="web", instance="02" }
http_request_duration_seconds

```
http_request_duration_seconds_count
{
  code="200",
  method="GET"
}
```

*add



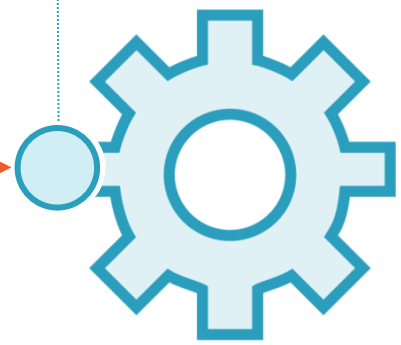
```
http_request_duration_seconds_count
{
  code="200",
  method="GET",
  job="web",
  instance="win2019",
  env="prod",
  os="windows"
}
```

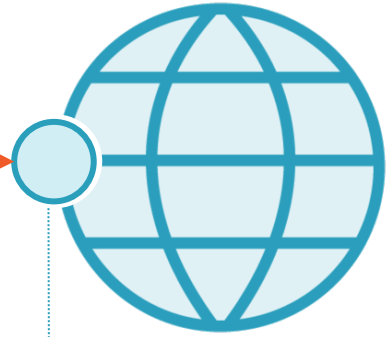

a_metric
{ os="windows" }
VM
ScaleSet
VM
VM
a_metric
{
__meta_azure_machine_os_type="windows",
__meta_azure_machine_name="vm01",
__meta_azure_machine_scale_set="vmss01"
}
*replace

```
http_request_duration_seconds
{ job="api", code="200" }
http_request_duration_seconds
{ job="web", code="200" }
```

```
http_request_duration_seconds
{ responseCode="200" }
```

*replace

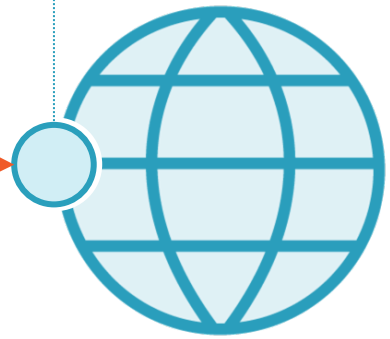




```
http_request_duration_seconds
{ code="200" }
```

```
http_request_duration_seconds
{
  code="200",
  method="GET",
  path="/",
  cacheHit="false"
}
```

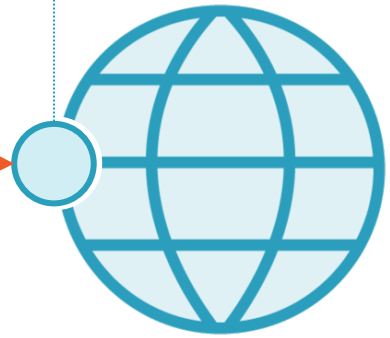
*labeldrop



```
http_request_duration_seconds_count
{
  code="200",
  method="GET",
  job="web",
  instance="win2019"}
```

```
method_duration_seconds
{
  class="UserRepository",
  method="GetUser"
}
```

*drop



Demo

Filtering and enriching labels

- Adding static labels
- Relabelling at target level
- Relabelling at metric level

Adding labels

Static configs

File service discovery

Target relabelling

Copy service labels

Filter target on labels

Metric relabelling

Edit or drop labels

Drop metrics

Summary

Running Prometheus

- Cross-platform binary
- Local storage
- Command-line + YAML config

Configuring Prometheus

- Scrape configs
- Target list - static or dynamic
- Platform integration for discovery

Target labels

- Relabelling to filter targets
- Replacing label values
- Relabelling to remodel metrics

Up Next:
Querying Metrics with PromQL