



Prometheus

Time series metrics, PromQL, Prometheus Operator,
Service Monitor, Grafana, Alert Manager

kloia in-a-nutshell:

Solutions in



DevOps/**DevSecOps**



Cloud/**Kubernetes**



Microservices



Test Automation

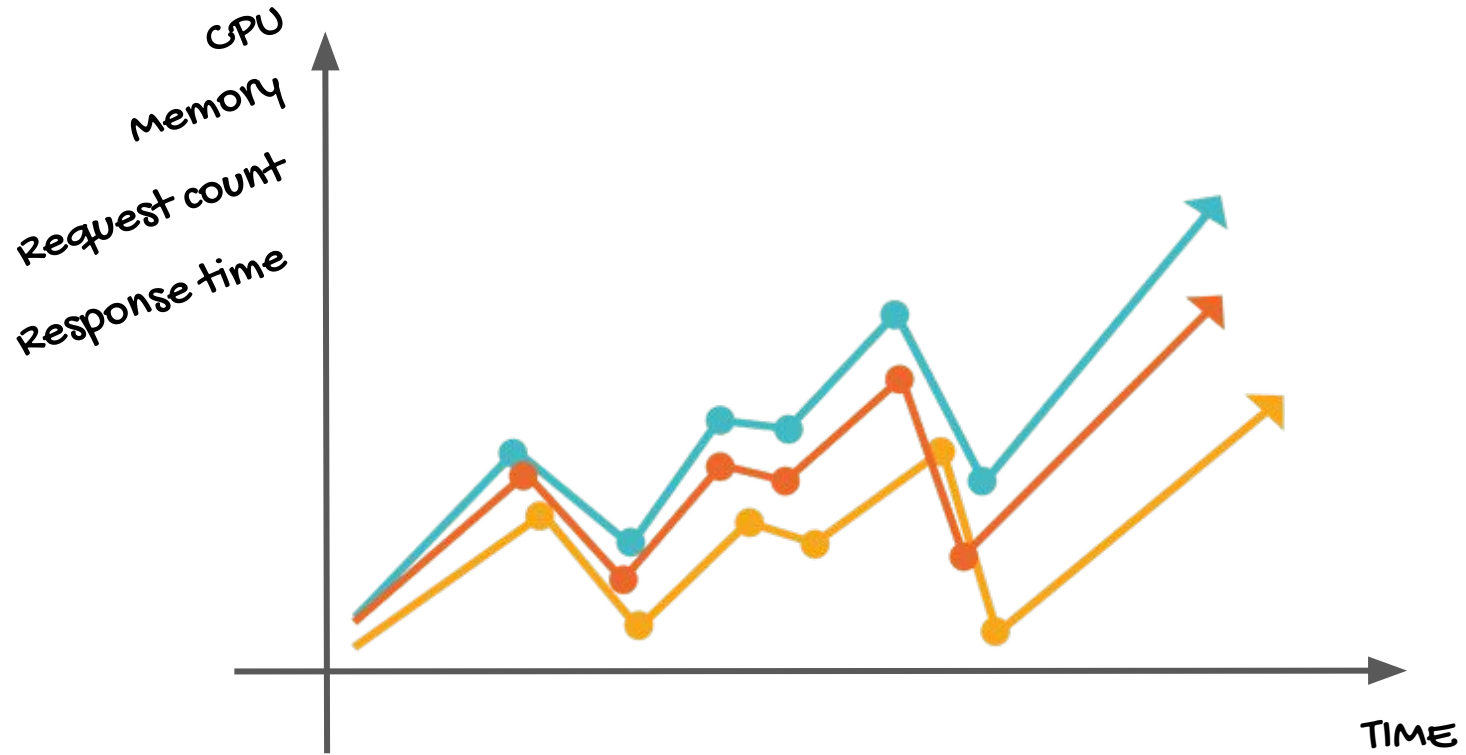


Customers from:

USA, United Kingdom, EU,
Tanzania, Dubai, Turkey

Within four nominated AWS
Partners globally for Modernization
Projects

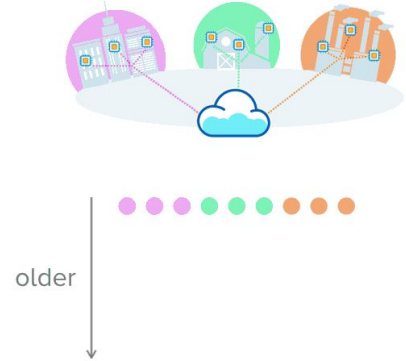
Several ongoing
modernization projects



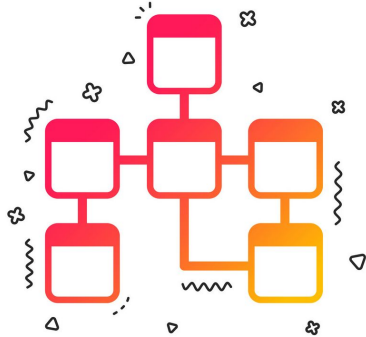
Neden Time Series Database/Metrik?

Time series vs. ne gerek var? İlişkisel database ile de çözeriz??

- Hızlı çalışır ve kolay scale olur.
- Uygulamanın durumu ile ilgili geriye dönük karşılaştırmaya imkan tanır.
- İstenilen durumlarda time series datadan alarmlar üretebilecek sistemler geliştirilmiştir.
- Anlık olarak sistemi izleyebileceğimiz arayüzler tasarlanmıştır.
- Özel geliştirilmiş sorgu dilleri ile istenilen metriklere kolay ulaşılır ve anormal durumların tespiti kolaylaşır.



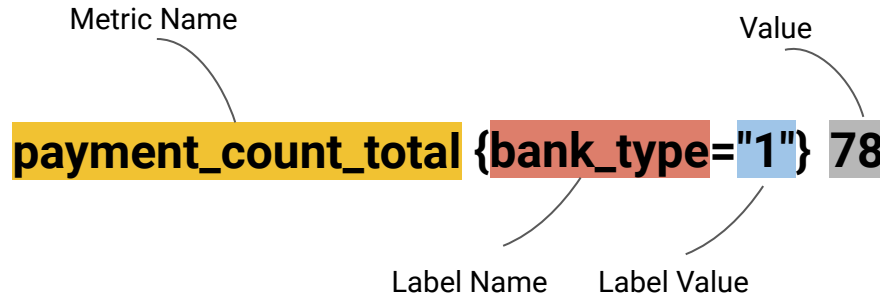
Son 1 Dakikada Kaç Adet Başarılı Ödeme İşlemi Gerçekleşti?



- Tablo üzerinde yapılacak sorgular çalışmakta olan sistemi aksatabilir.
- Kayıt sayısının artması zamanla sorgunun cevap süresini artırır.
- Verilerin görselleştirilmesi zordur.

```
SELECT * FROM Transactions  
WHERE   
BankType = 1 AND  
CreatedOn > DATEADD(minute, -1, GETDATE())
```

Son 1 Dakikada Kaç Adet Başarılı Ödeme İşlemi Gerçekleşti?



- Düzenli olarak bir değer üzerinden takip edilir. Çalışmakta olan sisteme bir etkisi yoktur.
- Her zaman tek değerdeki değişime bakıldığından kayıt sayısında hiçbir değişiklik olmaz. Sadece takip ettiğimiz değişkenin değeri değişecektir.
- Zaman ile ilişki, ilgili değer düzenli olarak okunup kaydedilerek sağlanır.

Prometheus

Açık kaynaklı, time series monitoring ve alarm yönetim aracıdır.

- Kendine has sorgulama dili vardır. (PromQL)
- Metrik adı, label(lar) ve bunlara karşılık gelen değerden oluşan bir metrik modeline sahiptir.
- Genellikle metrikleri belirli zaman aralıklarında kaynaklardan okumak üzere ayarlanır.
- Pushgateway özelliği ile kaynaklar tarafından metriklerin prometheus servera gönderilmesi de sağlanabilir.
- Metrikleri, grafana gibi birçok monitoring paneli üzerinde kullanılabilir.
- Topluluk tarafından kabul görmüş ve yaygın olarak kullanılmaktadır.

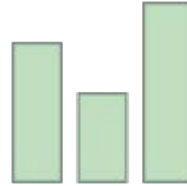
Prometheus Metrik Tipleri



COUNTER



GAUGE



HISTOGRAM



SUMMARY

Counter

- Sürekli artan değerleri göstermek için kullanılır. Toplam request sayısı, tamamlanan task sayısı veya error sayısı örnek verilebilir.
- **x_requests_total**, **x_task_completed_total** gibi isimlendirmeler kullanılır. (x: application name.)

```
paymentWithRecordedCardCounter = prometheus.NewCounterVec(  
    prometheus.CounterOpts{  
        Name: "session_prom_payment_with_recorded_card_total",  
        Help: "Number of payment completed with recorded card.",  
    }, []string{"bank_type"})
```

Gauge

- Counterdan farklı olarak artıp azalabilen değerleri göstermek için kullanılır. Anlık request sayısı, database üzerindeki anlık connection sayısı örnek verilebilir.
- İsimlendirme yapılırken değerın tipi sona yazılır. Örnek: **node_memory_usage_bytes**

```
payRequestsActiveGauge = prometheus.NewGaugeVec(  
    prometheus.GaugeOpts{  
        Name: "session_prom_pay_requests_active",  
        Help: "Number of pay requests processing.",  
    }, []string{"bank_type"})
```

Histogram

- Bir değeri gözlemlemek için kullanılır. İlgili değer için aralık verilerek, verilen aralıklara kaç kere denk geldiği saydırılabilir. Request duration ve ödeme tutarı örnek olarak verilebilir. Zamanla ilgili değerler saniye olarak tutulur.
- Değerlerin toplamını ve sayısını tutan 2 adet counter metrik tipi ile birlikte gösterilir.
- İsimlendirme yapılırken değerın tipi sona yazılır. Örnek: **http_request_duration_seconds**

```
paymentValueHistogram = prometheus.NewHistogram(  
    prometheus.HistogramOpts{  
        Name: "session_prom_payment_value_tl",  
        Help: "Histogram of received payment values (in TL)",  
        Buckets: []float64{20, 100, 200, 350, 500, 1000},  
    })
```

Summary

- Histograma benzer mantıkla çalışır. Elde edilen değerleri örnekler. Örneklemeyi, değerleri histogram olarak tutup sonrasında hesaplayarak da yapabiliriz.
- Histogram ile benzer mantıkta isimlendirme yapılır.

```
paymentDurationSummary = prometheus.NewSummaryVec(  
    prometheus.SummaryOpts{  
        Name:      "session_prom_payment_duration_seconds",  
        Help:      "Summary of payment duration seconds over last 10 minutes.",  
        Objectives: map[float64]float64{0.5: 0.05, 0.9: 0.01, 0.99: 0.001},  
    }, []string{"bank_type"})
```

PromQL

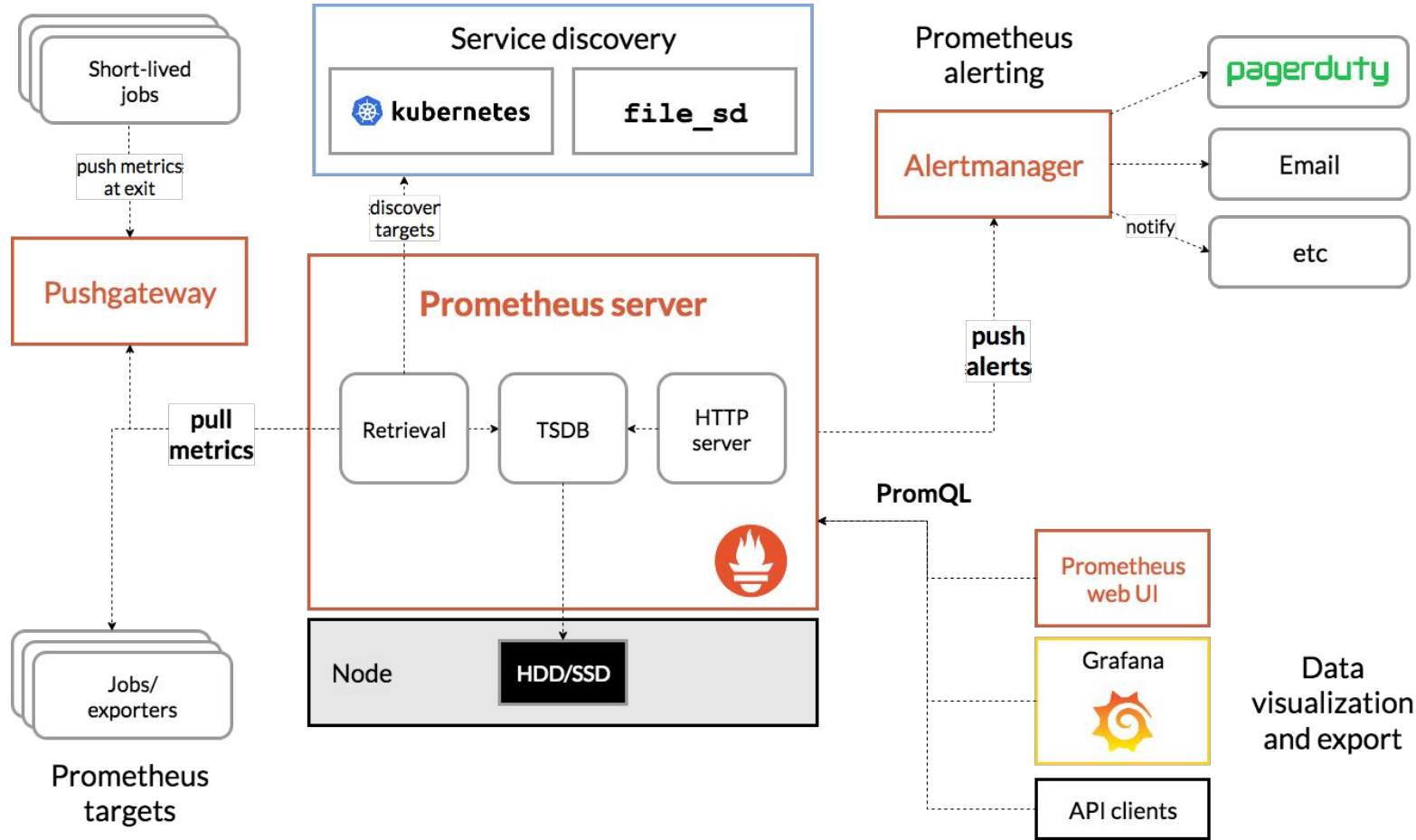
Elde edilen metriklerin sorgulanması için kullanılan prometheusa özel bir sorgulama dilidir.

- **+**, **-**, *****, **/**, **%**, **^** gibi aritmetik operatörleri destekler.
- **=**, **!=** gibi eşitlik içeren ya da **=~** gibi regex içeren ifadeler kullanılabilir.
`{job=~".*",method="get"}`
- **sum**, **rate**, **increase**, **round**, **avg**, **date** gibi birçok fonksiyona sahiptir.
- Dashboardlar ve Alarmlar PromQL ile yazılan sorgular üzerinden kurgulanır.

PromQL

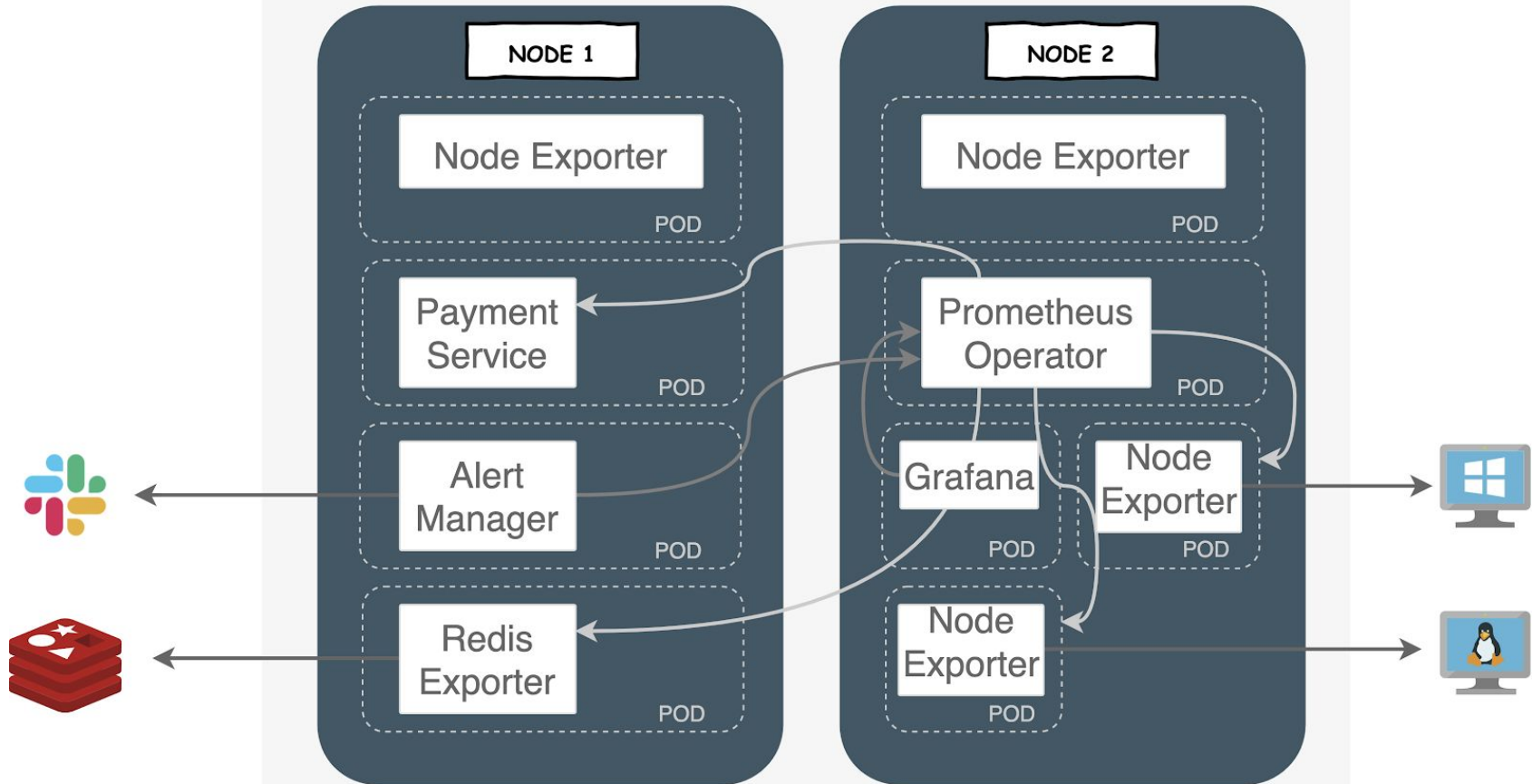
Son 5 dakika içerisinde A bankasından geçen işlemlerin yüzde kaçı başarılı oldu?

$$\frac{\text{increase}(\text{example_app_payment_count_total}\{\text{bank_type}=\text{"A"}, \text{success}=\text{"1"}\} [5\text{m}])}{\text{increase}(\text{example_app_payment_count_total}\{\text{bank_type}=\text{"A"}\} [5\text{m}])} * 100$$



Prometheus Operator & k8s

- Oluşturulan metrikleri toplaması için bir prometheus servera ihtiyaç vardır. Prometheus operator k8s üzerinde çalışan ve cluster metrikleri de dahil olmak üzere ayarladığımız tüm metrikleri toplayan uygulamadır.
- Helm reposu aracılığı ile grafana ve daha birçok özellikte birlikte paket olarak yüklenebilir.



Service Monitor

K8s içerisindeki uygulamaların metriklerini toplamak amacıyla prometheus operator tarafından oluşturulan CRD'dir. Metrikleri toplanacak servisler bu tanıma göre belirlenir.

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: session-prom
  namespace: monitoring
  labels:
    app: session-prom-monitor
    monitoring: prometheus-exporter
spec:
  selector:
    matchLabels:
      run: session-prom
  namespaceSelector:
    matchNames:
      - dev
  endpoints:
    - interval: 5s
      path: /metrics
```

Prometheus Dashboard


Prometheus server ile birlikte gelen bu dashboard ile,

- Metrikleri sorgulayabilir,
- Alarmları inceleyebilir,
- Metrik exporterlar ile prometheus server arasındaki bağlantı durumlarını

takip edebiliriz.

Prometheus Alerts Graph Status ▾ Help

☐ Enable query history

Expression (press Shift+Enter for newlines) 

Execute - insert metric at cursor · ▾

Graph Console

◀ Moment ▶

Element	Value
no data	

Remove Graph

Add Graph

Prometheus Rule

- **PrometheusRule**, prometheus operatore alarm ile ilgili ayarların iletilmesi için oluşturulan CRD'dir.

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  labels:
    monitoring: prometheus-exporter
  name: prom-session-alert-rules
  namespace: monitoring
spec:
  groups:
    - name: prom-session
      rules:
        - alert: PaymentTransactionLow
          annotations:
            message: 'Bank {{ $Labels.bank_type }} payment transaction count
lower than usual for 1 minute interval. Current {{ $value }}'
          expr:
round(increase(session_prom_payment_duration_seconds_count[1m])) <
20
          labels:
            severity: critical
          for: 1m
```

Alert Manager

Alert Manager ise tetiklenen alarmları belirlenen şekillerde mail ya da slack gibi kanallar ile gönderme aracıdır.

[Alertmanager](#) [Alerts](#) [Silences](#) [Status](#) [Help](#) [New Silence](#)

Filter

Group

Receiver: All ☐ Silenced ☐ Inhibited

[+](#) [Silence](#)

Custom matcher, e.g. `env="production"`

[+ Expand all groups](#)

[+](#)

Not grouped

1 alert

[+](#)

Not grouped

1 alert

[+](#)

Not grouped

41 alerts

[+](#)

Not grouped

1 alert

Grafana Dashboard

Prometheus veya başka bir time series database ile export edip depoladığımız metrikleri görsel açıdan zengin ve daha anlamlı hale getirebileceğimiz open source bir monitoring aracıdır.



Dinlediğiniz için teşekkürler.

Soru & Cevap



Advanced
Consulting
Partner

DevOps Competency

Microsoft Workloads
Competency

Solution Provider

- 30+ consultants
- DevOps, Microservices, Test
- %75 kloians working remotely
- Quarterly Summits
- Offices in London & İstanbul
- Customers from US, UK, EU
- Our Event Sourcing Framework: eventapis.kloia.com
- Send your CV to career@kloia.com



CLOUD NATIVE
COMPUTING FOUNDATION
Silver Member

kloia is a solution provider that
moves you onto the new-era in
DevOps, Cloud and architecture.



kloia.co.uk



kloia.com



blog.kloia.com



[@kloia_com](https://twitter.com/kloia_com)



33 Queen St, London EC4R
1AP, United Kingdom



AND Binası 10-12, 34752
Kozyatagi/Istanbul, Turkey