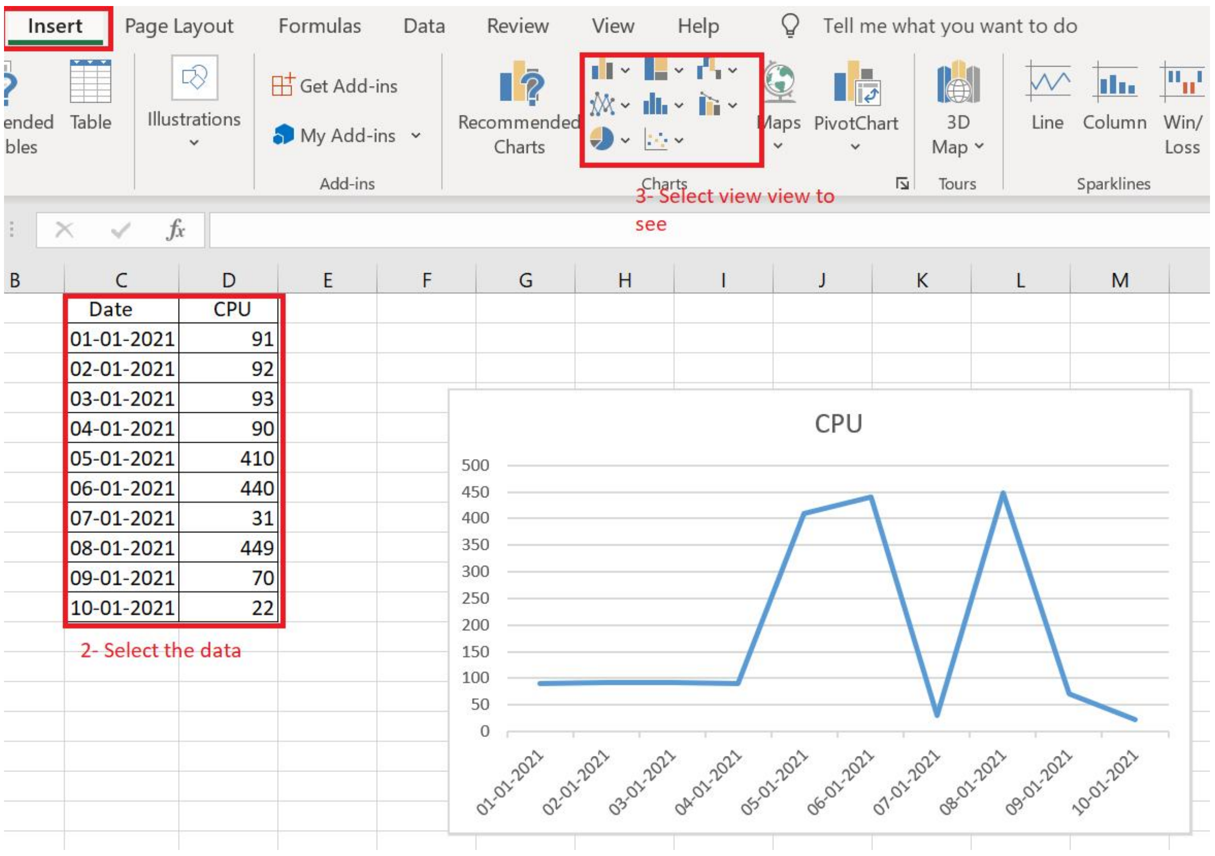
Monitoring and Logging in Kubernetes

- Metrics vs Logs:
  - Metric: A series of numbers over a period of time
  - Logs: Log represents some activity or information occurred in the application/system. Logs have data in the form of text (majorly) in some format



Date	CPU
01-01-2021	91
02-01-2021	92
03-01-2021	93
04-01-2021	90
05-01-2021	410
06-01-2021	440
07-01-2021	31
08-01-2021	449
09-01-2021	70
10-01-2021	22

- Monitoring Techniques: (Class 1, 2, 3 and 4 monitoring)
  - Black-box monitoring focuses on monitoring from outside of an application and is what has been used traditionally when monitoring systems for components like CPU, memory, storage and so on. This can be still useful for monitoring at infrastructure level, but it lacks insights and context to how application is operating.
  - White-box monitoring focuses on details in the context of application state, such as total HTTP requests, number of 500 error, latency of request and so on. With this monitoring we can understand why of our system state.

Monitoring Patterns

- Kubernetes is much more dynamic and transient, and we need to change the way how we monitor these environments
- There are couple of different monitoring patterns to focus on when monitoring distributed systems
  - The USE method (Bredan Gregg) :
    - Focuses on
      - U – Utilization
      - S – Saturation
      - E – Errors

        - This method is focused on infra monitoring. The USE Method can be described as "For every resource, check utilization, saturation and error rates"
        - This method helps in quickly identifying resource contstraints and error rates of systems
    - The RED Method (Tom Wilke):
        - Focuses on
            - R – Rate
            - E – Errors
            - D – Duration
        - This philosophy was taken from Google Four Golden signals
            - Latency (How long it takes to serve a request)
            - Traffic (how much demand is place on system)
            - Error (rate of requests that are failing)
            - Saturation (How utilized your service)

Kubernetes Metrics Overview

- Let's look at what components we should be monitoring in k8s cluster.
- In K8s we need to monitor
    - Control-Plane Components
        - API Server
        - etcd
        - Scheduler
        - Controller
    - Worker Node Components
        - Kubelet
        - Container Runtime
        - Kube-proxy
        - Kube-dns
        - Pods
- Kubernetes exposes the above metrics in variety of ways, so lets look at different components that we can use to collect metrics in your cluster
    - cAdvisor:
        - Container Advisor or cAdvisor is an open source project that collects resources and metrics for containers running on a node. cAdvisor is built into the k8s kubelet, which runs on every node in Cluster
        - It collects memory and CPU metrics using cgroup.
        - It also collects disk metrics through statfs (which is built into the Linux Kernel)
        - We should be considering cAdvisor as source of truth for all container metrics
    - Metrics Server: (Link: https://github.com/kubernetes-sigs/metrics-server)
        - The k8s metrics server and Metrics Server API are replacement for deprecated Heapster.
        - There are two aspects to understand in Metrics Server API and metrics server
            - The implementation of Resource Metrics API is the metrics server. This metrics server gathers resource metrics such as CPU and memory. It gathers these metrics from kubelet's API and stores in memory. K8s uses these resource metrics in the

    - scheduler,Horizontal Pod AutoScaler (HPA) and Vertical Pod AutoScaler (VPA)
    - The Custom Metrics API allows monitoring systems to collect arbitrary metrics. This allows monitoring solutions to build custom adapters that will allow for extending outside the core resource metrics. For example Prometheus built one of the first custom metrics adapters, which allows us to use the HPA based on a custom metric.
  - kube-state-metrics: (Link https://github.com/kubernetes/kube-state-metrics)
    - This is k8s add-on that monitors the object stored in k8s.
    - Where cAdvisor and metrics server are used to provide the detailed metrics on resource usage, kube-state-metrics is focused on identifying conditions on k8s objects deployed on the cluster
    - Following are some questions that kube-state-metrics can answer
      - Pods
        - How many Pods are deployed to the cluster?
        - How many Pods are in Pending State?
        - Are there enough resoruces to serve a pod request?
      - Deployments
        - How many pods are in running state vs desired state?
        - How many replicas are available?
        - What deployements have been update?
      - Nodes:
        - Whats the status of the Worker Nodes?
        - WHat are allocatable CPU cores in my cluster?
        - Are there any nodes that are unschedulable?

Monitoring Tools

- There are many monitoring tools that can be integrated with K8s. Following are few popular tools
  - Prometheus
  - Influx DB
  - Datadog
  - Sysdig
  - Cloud Provider Tools:
    - GCP Stackdriver
    - Microsoft Azure Monitor for containers
    - AWS Container Insights

Logging Overview

- In k8s cluster, there are multiple components to log
- Following is a list of components from which you should be collecting metrics
  - Node logs
  - K8s Control-Plane logs
    - API Server
    - Controller Manager
    - Scheduler
  - K8s audit logs
  - Applciation Container logs

Tools for Logging

- Some of more popular tools with k8s integration
    - Elastic Stack
    - Data dog
    - Sumo Logic
    - Sys dig
    - Cloud Provider Services
        - GCP Stack Driver
        - Azure Monitor for Containers
        - AWS Cloud Watch