**1. Introduction to Sensu**

* **What is Sensu?**
* Sensu is an **open-source monitoring and observability tool** that helps teams keep track of their systems, applications, services, and infrastructure.  
  Think of it as a **watchdog** that continuously checks if everything in your environment is working as expected.
* **Category**: Infrastructure Monitoring & Observability.
* **Main Purpose**:
  + Monitor servers, containers, cloud resources, and applications.
  + Collect metrics and events.
  + Send alerts when issues occur.
  + Integrate with other tools (like Slack, PagerDuty, Prometheus, Grafana).

**2. Why Sensu?**

* Traditional tools (like Nagios) are powerful but **hard to scale** in modern cloud environments.
* Sensu was designed to handle **dynamic, cloud-native infrastructures** (VMs, containers, Kubernetes).
* It supports **monitoring as code** → monitoring checks and configuration are written as code (YAML/JSON).

**3. Key Features of Sensu**

1. **Event-Driven Monitoring**
   * Everything in Sensu is based on **events** (like system states, alerts, or metrics).
   * An event can trigger actions (alert, log, remediation).
2. **Service Checks**
   * Similar to Nagios checks.
   * Can run scripts/plugins to verify if a service is running, a port is open, etc.
3. **Metrics Collection**
   * Collects performance data (CPU, memory, disk, response times).
   * Can export metrics to time-series databases (InfluxDB, Prometheus).
4. **Alerting & Notifications**
   * Sends alerts to email, Slack, PagerDuty, VictorOps, etc.
   * Supports rules and filters to avoid alert fatigue.
5. **Integrations**
   * Works well with Prometheus, Grafana, InfluxDB, and other observability tools.
6. **Scalable & Cloud-Native**
   * Designed for Kubernetes and dynamic infrastructures.
   * Supports auto-discovery of services.

**4. How Sensu Works (Simple Flow)**

1. **Agent** runs on each monitored host (server, container, VM).
2. Agents run **checks** → (e.g., “Is CPU usage too high?”).
3. Results of checks are sent as **events** to the **Sensu backend**.
4. Sensu processes the events:
   * Store in a database
   * Forward metrics to Grafana/Prometheus
   * Trigger alerts if thresholds are crossed
5. Teams act on alerts and dashboards.

**5. Benefits of Sensu**

✅ Flexible → can monitor traditional servers + modern cloud setups.  
✅ Supports **monitoring as code** → easy automation & version control.  
✅ Scales well in Kubernetes/microservices environments.  
✅ Strong **integration ecosystem** (Grafana, Prometheus, Slack, etc.).  
✅ Open-source + Enterprise options.

**6. Who Uses Sensu?**

* **DevOps teams** → monitor CI/CD pipelines.
* **SREs (Site Reliability Engineers)** → ensure system uptime.
* **IT Operations** → server and network monitoring.
* **Cloud-native teams** → Kubernetes and container observability.

**7. Example Use Cases**

* Monitor if a **web service** is up (HTTP check).
* Track **Kubernetes pods** health.
* Alert if **CPU > 90%** on a VM.
* Collect metrics and visualize them in **Grafana**.
* Auto-remediate → restart a service if it goes down.