AP/ITEC 2210 3.0 A: System Administration Fall 2023

Instructor: Jamon Camisso
ITEC 2210 Chat: mattermost.itec2210.ca
Email: jamon@yorku.ca
Website: https://eclass.yorku.ca/

Date/Time: Wednesday, 19:00-22:00
Location: Zoom / ACE 003
Office hours: Via Mattermost any time

Class 9 - Monitoring

- Announcements
- Course Evaluations:
  - https://courseevaluations.yorku.ca/
- Final Exam:
  - 11 Dec, 19:00, room TBD
  - :(

Class 9 - Monitoring

- Readings:
  - SRE - Monitoring Distributed Systems https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems/
  - PSNA Chapter 38
- Next week:
  - PSNA Chapter 4

Class 9 - Monitoring

- Principles of monitoring:

  “Every page that happens today distracts a human from improving the system for tomorrow” SRE, p.64

Class 9 - Monitoring

- SRE Principles of monitoring:
  - Design monitoring with KISS principle in mind
    - Many small components versus a monolithic system
  - Pages should require human intervention
    - Avoids alert fatigue
  - Focus alerting on symptoms, not causes
    - Counterintuitive, but reduces false-positives

Class 9 - Monitoring

- Types of monitoring
  - **PSNA** defines two kinds of monitoring:
    - Real-time
    - Historical
  - **SRE** describes two other types:
    - White-box
    - Black-box

Class 9 - Monitoring

- PSNA Types of monitoring
  - **Real time -** consists of two components
    - Detects failures when they happen
    - Alerts someone to the failure
  - **Historical -** consists of two components
    - Data collection system (remote, local, push, pull)
    - Display, visualization, or dashboard component

Class 9 - Monitoring

- SRE Types of monitoring
  - **White-box -** preemptive detection of problems
    - Relies on instrumentation, logs, inspecting systems
    - Can predict imminent service outage/degradation
    - End to end knowledge of a system's internals
  - **Black-box** - active, unexpected problems
    - SOMETHING IS BROKEN RIGHT NOW!
    - Reduces alert fatigue - only actual live issues alert
    - No internal knowledge of a system

Class 9 - Monitoring

- **Historical Monitoring**
  - Collects and stores monitoring data over time
  - Used in capacity planning
  - Helpful with past incident analysis and detection
  - SLA conformance - when outages happen, how long?

Class 9 - Monitoring

- **Historical Monitoring**
  - Retention periods and compression are important
    - Billing data (usage based billing) likely needs to be kept in some form for as long as a customer exists
    - Internal performance data can be rotated out over time until it is deleted (usually moving window)

Class 9 - Monitoring

- **Historical Monitoring**
  - Collection mechanisms matter!
    - A polling system will only be able to handle so many systems per poll interval
    - A push architecture can overload the central monitoring system

Class 9 - Monitoring

- **Historical Monitoring**
  - Though disk space is cheap, **storage** and **expiring** data is important too
    - **Summarize** data on various intervals - probably don't need per minute data 2 weeks from now
    - Use things like **RRDtool** (round robin database) to **rotate/expire** data on a rolling window basis

Class 9 - Monitoring

- **Historical Monitoring**
  - Visualization using graphs e.g. Prometheus & Grafana
  - Easy to spot trends or anomalies
  - https://play.grafana.org
  - Also useful to circulate to management, peers, customers when troubleshooting, justifying new gear

Class 9 - Monitoring

- **Real-time Monitoring**
  - A check should be designed to detect failures
    - Something is broken, fix it is the criteria for a check
  - Usually minimal storage requirements, since each check just gets replaced with the results from the next run
  - Should use standard, well-understood protocols, for push and pull checks

Class 9 - Monitoring

- **Real-time Monitoring**
  - Data can come from log files, SNMP, HTTP checks etc.
  - Alerts should be generated directly from data
    - Don’t parse or format data, just alert if it exists, or contains unknown data, or known errors
  - Alerts should use a variety of mechanisms - email, pager/app, phone, SMS, tickets

Class 9 - Monitoring

- **Real-time Monitoring**
  - Real-time like Nagios/Icinga2, Zabbix, Prometheus, Datadog, Sensu should check for:
    - Service availability - is it up and responding?
    - Service capacity - is it about to run out of x?
    - Service flapping (intermittent issues)

Class 9 - Monitoring

- **Log processing**
  - Logs can (and should) be sent to a central logging tool
  - ELK (Elasticsearch, Logstash, Kibana) is common
  - An agent on a system forwards logs to central tool
  - Various alert criteria can be applied, real-time and historical based on resource usage

Class 9 - Monitoring

- **Alerting**
  - Alerting & monitoring should be separate in that if either is down, the other still works
  - e.g email can be down for alerts, but monitoring still works, and other alerting mechanisms do too
  - Be careful with alerts that contain sensitive data? NO PASSWORDS! 3rd parties can be listening in.

Class 9 - Monitoring

- **Alerting policies**
  - Who gets pages? NOC? SA? Managers?
  - How often does something realert? 5, 15, 30 minutes?
  - When do you escalate if something isn't resolved?
  - What happens if a page is missed? Who gets it?
  - How severe is the problem? What is broken?
  - E.g. Disk capacity probably shouldn't escalate to CIO
    - Need some ability to group or target alerts

Class 9 - Monitoring

- **Alerting policies**
  - Acknowledging an alert is crucial to informing others it is being worked on, and to avoid re-alert interruptions
  - However, be careful because ACK will turn alerting off until it is un-acknowledged
  - Maintenance mode or Downtime windows are essential for deployments, or working on problems

Class 9 - Monitoring

- **Active Monitoring**
  - Personal opinion: just don't
  - Better to spend time fixing whatever the problem is
  - If you must, limit privileges of automation tool, be sure to generate a ticket or log of every automatic action
  - **Monit** is one such tool

Class 9 - Monitoring

- **Scaling issues**
  - Large networks can become clogged near monitoring endpoints, with thousands of checks a minute
  - Some systems summarize data and collect that
  - Other option is multiple smaller monitoring systems
    - Can keep traffic contained, manageable configuration, escalation, and alerting

Class 9 - Monitoring

- **Coordination/Prioritization**
  - Prevent multiple people from working on an alert - wasted effort, or worse, conflicting solutions
  - One option: create a dedicated alert duty role
  - Or have whomever is on ticket triage handle alert triage as well, and process escalations appropriately

Class 9 - Monitoring

- **Centralization & Documentation**
  - Keep copies of configuration and checks in a centralized version control system
  - Ensure any SA can add a check, after review from any other SA
  - Even with multiple different monitoring systems, use a branch in the VCS to keep things centralized

Class 9 - Monitoring

- **Centralization & Documentation**
  - Try to make checks and alerts as self-documenting as possible
    - E.g. ‘nexus-5000-4-port48 spanning tree is disabled’
    - Not, ‘spanning tree is disabled on core switch’
  - Ensure docs for adding checks and a playbook for resolving checks exists in a central location

Class 9 - Monitoring

- **Pervasive Monitoring**
  - Add monitoring components to a deployment during server/VM provisioning
  - Ensure it is built in from the start, and that checks are tailored to the service being deployed
  - Rely on inventory system to detect and check systems - automation will save troubleshooting new changes

Class 9 - Monitoring

- **End to End testing**
  - Actually test systems end to end
  - Have test users with secure login credentials, test systems using them: email, HTTP login, ordering, etc.
  - Check transaction times, that data exists in a database after a test login, and check remote APIs that may be integrated - many times an external outage will cause slow downs or application downtime

Class 9 - Monitoring

- **Meta-monitoring**
  - Monitor your monitoring
  - Do a periodic end to end smoke test of monitoring
  - E.g. check for a file generated by cron on some interval, alert if the test does **not** complete or the file is out of date, doesn't exist etc.