Debugging cluster issues as on-call SRE

- Pravar Agrawal

Agenda

- Introduction to Reliability Engineering
- Understanding role of an on-call engineer
- Identifying some commonly occurring cluster-level issues
- Approach to debugging
- Automation to the rescue
- What to do if you are a beginner

whoami

- Senior Engineer @IBM (IKS)
- Co-host for Bangalore SRE Meetup Group - https://meetup.com/sre-bangalore/
- Writes @ pravarag dot com

@realpravarag

@pravarag on K8s

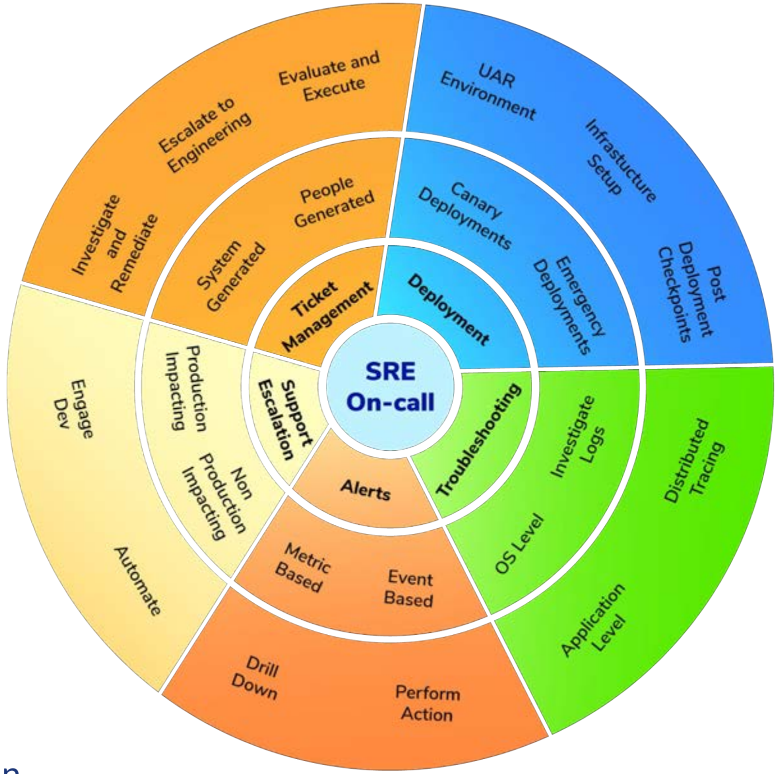
@pravarag

Introduction to SRE

- An approach to IT operations using different tools to solve problems, manage systems and automate operations tasks.
- Valuable practice for creating scalable and reliable systems
- SRE practitioners ensure balance between releasing new features and reliability for users.
- Being on-call means, you are available for a set period of time and ready to respond to production incidents during that time with urgency.

Understanding on-call process

- Different companies may have their own implementations of on-call process
- Main aim is to support the production 24x7 by incident management following few rules:
  - Acknowledge and verify the alert
  - Analyze the impact
  - Communicate
  - Corrective Action or Fix
- Famous tools for incident management: PagerDuty, Jira, OpsGenie, ServiceNow etc.



Source: squadcast-on-call-rotation

Some common cluster issues

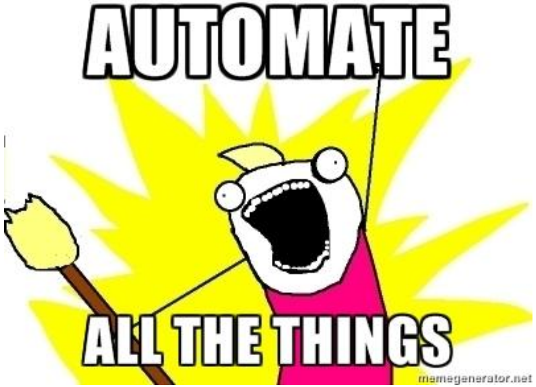
- An environment comprising of a single or multiple Kubernetes clusters in production
- Issues related to services running on node. Is it possible to manually ssh over those to check?
- Multiple pods stuck in Pending or Terminating state.
- API endpoints down or not reachable
- 1 or 2 Etcd pods not available out of HA
- Issuing reloads of worker nodes
- Disk reaching capacity for a worker node
- Health checks failure

Approach to Debugging

- There are no Golden rules, but there are right ways to do it.
- Analyze the error message received - lower the blast radius, better it is.
- Utilizing monitoring tools like Prometheus, LogDNA to look at the last recorded state of application.
- If it's K8s related, get access to the cluster and try to list out status of master components, etcd and other core components.
- If it's a widely impacting issue, try to isolate the service by restricting it's usage throughout.

DEBUGGING
I DON'T KNOW WHERE YOU ARE. I DON'T KNOW HOW YOU WORK. BUT I WILL FIND YOU. AND
I WILL FIX YOU

Automation to the rescue?



- Reducing the time to respond and get infrastructure statistics at the earliest.
- Automate to:
  - Get cluster statistics
  - Run real time commands, handle node reboots or restart core services
  - Query historical data to find patterns in occurrence of different issues.
  - Schedule clean-up jobs

Shades of automation

- Meaningful and well documented Runbooks for on-call SREs
- Chatops
- Bots - Slack, Teams, mattermost
    - https://botkube.io/
- Purposely curated scripts running as K8s resources
- AI based analysers to gather much detailed info from the cluster
- Integrations with existing monitoring tools to extend their capabilities
- Curated dashboards to get a better view of what’s happening inside the infra

Advice for beginners

- More the exposure, more seasoned you'll get in handling different situations
- Broader understanding of the entire architecture and infrastructure involved.
- Keep runbooks handy to deal with different issues, errors, warnings etc.
- Analyze historical or recent issues and alerts that have caused different outages. This could prove as a great learning.
- *If it auto-resolves, doesn't always mean there's nothing wrong.*

Thank You!