CloudOps is the "formalization of best practices and procedures that allow cloud-based platforms, and applications and data that live there, to function well over a long duration of time."

With containers, complex microservices and serverless functions becoming more commonplace, teams need to find ways to move faster without compromising stability and security.

Manage the operations from various stakeholders' perspective, role, identity and purpose. Logging, monitoring and performance management is an old DevOps stuff. What new in cloud.

Cloud is a dynamic world and not static, hence Ops are dynamic and needs to be continuous

"minimal viable cloudops toolset," or MVCOT

How do you find this MVCOT nirvana? I would suggest the following:

- Define the current, short-term future, and long-term future state of the systems that will be under management.
- Define the runbooks for each system. These will explain how each system needs to be managed to provide optimal service levels and minimize outages
- Now it's time to select the right tools to make up your MVCOT

Each system, product, app and service uploaded added needs a dynamic entry in a master inventory db. This db needs to be intelligent to verify the dependencies, updates and conflicting dll, codebase etc. This should alert the user in case of issues.

This will part of the Orchestrator or the cloud monitor that will host agents, instrument and do health checks to ensure integrity is not compromised at least in prod machines.

Cloud orchestration in a multi-cloud and hybrid-cloud environment is not an easy task. There is a need for cross-platform, cross-operating system communications and a need for robust design to get the required instrumentation, hooks, and passing on messages at various levels is needed. Orchestrations typically address the upper layers of the OSI framework like L7. However, there will be a need for calls at the OS level apart from the user space calls. It becomes more complex for various level calls at containers, OS shell, multiple virtual stacks, and translation of addresses; there is no easier way to get an easier cross-talk.

There is no room for errors like segment failure, null pointer exception, RPC failure, out of memory exceptions, & non reachable code etc etc . System integrity and SLA agreements need to address such issues to honor their assurances

Few tools:

1. Google Cloud's operations suite (formerly Stackdriver)
2. Microsoft Azure Automation.
3. Puppet Bolt

4. BMC Multi-cloud Management
5. Morpheus
6. CloudHealth
7. Chef
8. Ansible
9. OpenStack
10. Amazon Cloudwatch
11. AppDynamics
12. BMC TrueSight Pulse
13. DX Infrastructure Manager
14. New Relic
15. vRealize Hyperic
16. Solarwinds
17. ExoPrise SaaS monitoring
18. Retrace
19. Apache CloudStack
20. ManageIQ
21. Cloudify