Kubernetes (K8s) is an open-source platform for managing containerized workloads and services. It acts as a portability tool, reusable wherever you are deploying. Kubernetes architecture leverages containerisation to modularise apps and services which may require many different processes. K8s runs on top of bare metal, private/public cloud and OpenStack resources such as compute, memory storage and network can be configured as Kubernetes Nodes. Kubernetes demands an automated approach and an automated approach means IaC.

Kubernetes competitiveness strength is a powerful orchestration DevOp tool, suitable for massive scale deployments on different environments due to containers configured as a clusters. Kubernetes can be automated as an in-cluster CI/CD pipeline. A CI pipeline utilises a docker image and creates a container image pushing it to the container registry. Inside a container image are all the necessary executables, binary code, libraries, and configuration files. A Kubernetes deployment can be a cluster or multicluster which requires a Docker/ Github repository from which to pull your images. In DevOps K8s supports zero downtime, rollback, scaling, and container self-healing. It is a framework on which many tools can be integrated such as continuous delivery (ArgoCD, Spinnaker), network policy enforcement (Calico), management portal (Skooner), performance metrics (Prometheus, Grafana), logging (Fluentbit), DNS server (CoreDNS) and all-in-one CI/CD tools (GitLab CI/CD, Jenkins, GitLab CI/CD).

Kubernetes is on the upswing as Gartner predicts a rapid increase. Yet the container orchestration tool faces security threats from multiple aspects, namely its different layers. According to (Y.Yang, W.Shen, B.Ruan, W.Liu and K.Ren 2021) A malicious container can exploit kernel vulnerabilities in lower layer kernel. Another potential intrusion is the API server does not validate YAML files to the extent a DoS attack can be crafted. Defects in orchestration tools give the attacker chances to control the whole container system e.g., crypto-mining related malicious images on the Docker hub. Moreover (J. Flora, M. Teixeira and N. Antunes 2023) suggest security vulnerabilities of container images, privilege escalation and poor container isolation that can comprise organisation pipelines. Thus, monitoring agents and tool detection alarms allow users to interact via dashboards to strengthen security. I believe a strong knowledge of API configuration is fundamental to security and staying up to date with newer k8s security tools released.

1. Y. Yang, W. Shen, B. Ruan, W. Liu and K. Ren, "Security Challenges in the Container Cloud," 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Atlanta, GA, USA, 2021, pp. 137-145, doi: 10.1109/TPSISA52974.2021.00016.
2. J. Flora, M. Teixeira and N. Antunes, "µDetector: Automated Intrusion Detection for Microservices," 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Taipa, Macao, 2023, pp. 748-752, doi: 10.1109/SANER56733.2023.00084.
3. James Freeman; Jesse Keating, *Mastering Ansible: Automate configuration management and overcome deployment challenges with Ansible* , Packt Publishing, 2021.

Continuous integration benefits developers code integrated into the main branch of their software project. Hosting a build server such as Jenkins pulls code changes, and performs automated tests using multiple integrated tools, integrates them into a main branch, and creates a new version of software artifacts

Continuous deployment tools refer to automatically releasing a developer's changes from the registry to the staging or production destination environment.

continuous testing (CT),

continuous deployment,

continuous monitoring, continuous feedback, and continuous operations.

develop, build, test and deploy

https://securitycafe.ro/2023/02/27/a-complete-kubernetes-config-review-methodology/

design expands containerisation

Secure set up, high avail key value store manage user tenancy, isolate traffic for different users.

within The Software Development Life Cycle (SDLC). Similar to a VM, a container has its own filesystem, share of CPU, memory, process space. As they are decoupled from the underlying infrastructure, they are portable across clouds and OS distributions.

In DevOps application K8s can leverage container images for decoupling applications at release stage to compile, generate code, package modules independently rather than deployment stage. This further facilitates easier code maintenance and greater agility for change.

<https://ieeexplore.ieee.org/document/9936478>

https://www.youtube.com/watch?v=q6BfRiI4q0w

https://securitycafe.ro/2023/02/27/a-complete-kubernetes-config-review-methodology/#wrap-up

https://www.aquasec.com/cloud-native-academy/kubernetes-101/kubernetes-alternatives/

<https://www.plutora.com/devops-at-scale/pipeline>

You can choose from categories like version control, containerization, orchestration, automation, monitoring, and more.

**Brief description of its purpose**

(James Freeman; Jesse Keating 2021)

Ansible is designed as an opensource multi-tier orchestration, configuration and deployment functional DevOps tool. It can execute configurations on target infrasture resources and describe the desired state, such as, a file or directory exists, package installation, a service is running. Ansible is written in Python and configured in YAML.

Ansible supports the CI/CD pipeline in terms of integration with Jenkins, GitLab, or CircleCI to build environments. Such scenario allows Jenkins deploys a staging environment on every code change. The pipeline can trigger Ansible playbooks based on events like code commits, scheduled intervals, or manual triggers to pass or fail a build. Sensitive credentials, API keys, and SSH keys can be injected into pipleline steps at runtime. If the deploy job succeeds, it runs the same deploy playbook against production infrastructure.

An Ansible playbooks incorporates a sequence of tasks for execution in the order they appear which fails on errors, thus, failing fast protects some parts of InfrastruX continuing to execute. Furthermore, a service may not react well by shutting down all running processes and reboots. Playbooks can use variables to create a separate data class with the fields you want **to serialize.** Playbooks lend clustering restarts at end of bug fixes, new features and security updates can reduce interruption as apposed to progression restarts.

In terms of IAC Cloud automation Ansible can boot, configure, manage servers and manage facts, actions and volumes for all virtual machines of a resource group, including start/stopping inventory for specific images. Moreover, code can be deployed on Docker containers using ansible playbooks and the process automated using jenkins.

if you wish to use it to roll out systems as part of a Continuous Integration/Continuous Delivery pipeline.

Ansible describes states of resources declaratively. validate basic things of your infrastructure