Observability and Monitoring

Observability and Monitoring tools track the performance, health, and availability of applications and infrastructure, enabling teams to detect and diagnose issues to ensure systems run efficiently.

- **Prometheus**: An open-source monitoring solution that collects and stores metrics as time series data, featuring a powerful query language for data analysis. It is commonly used in conjunction with Grafana for visualising the metrics
- **Grafana**: A versatile open-source analytics and visualisation web application that supports various data sources, including Prometheus. It is renowned for its ability to create detailed dashboards that provide insights into metrics and logs
- **Fluentd**: An open-source data collector designed for unified logging layer, allowing the integration of data collection and consumption to enhance data understanding. Its vast plugin ecosystem supports data connections with multiple sources and destinations
- **KubeCost**: Focuses on cost management for Kubernetes, enabling teams to monitor, analyse, and optimise their Kubernetes spending efficiently

Service Mesh

Service Meshes offer a robust infrastructure layer that facilitates secure, reliable service-to-service communication within microservices architectures, enhancing observability, reliability, and security.

- **Cilium**: Utilises eBPF technology to provide advanced security features, network connectivity, and load balancing for Kubernetes clusters. It's integral for securing network communication and enforcing security policies
- **Linkerd**: A lightweight, high-performance service mesh that simplifies service-to-service communication with built-in observability, tracing and security, requiring minimal configuration
- **Istio**: Offers a comprehensive service mesh solution, enabling fine-grained control over service communication with features like secure service-to-service communication, traffic management, and extensive observability. Istio uses Envoy as its default service proxy
- **Envoy**: Designed for modern cloud-native applications, this edge and service proxy is the foundation for various service mesh implementations, including Istio, providing dynamic service discovery, load balancing, and TLS termination

Container Runtime and Execution

This section covers tools essential for container management, highlighting the distinction between container runtimes and Kubernetes management solutions.

- **Docker**: The most popular container platform, providing the ability to package software into standardised units for development, shipment, and deployment
- **Kata Containers**: Combines the speed of containers with the security of virtual machines, offering an isolated environment for container execution
- **containerd**: An industry-standard runtime focused on simplicity, robustness, and portability, used by Docker and Kubernetes
- **CRI-O**: Tailored for Kubernetes, providing a lightweight container runtime that fully complies with the Kubernetes Container Runtime Interface
- **runc**: A CLI tool for spawning and running containers according to the Open Container Initiative (OCI) specification
- **gVisor**: An open-source container runtime, designed to provide a sandboxed environment for running containers, enhancing security by isolating the workload from the host kernel

Kubernetes Management and Distributions

Tools and platforms designed to simplify the deployment, management, and operation of Kubernetes clusters.

- k3s: A lightweight, easy-to-install Kubernetes distribution, ideal for edge computing, IoT, and CI/CD environments due to its minimal resource requirements
- minikube: Enables the running of a Kubernetes cluster on personal computers, ideal for development and testing purposes
- EKS (Amazon Elastic Kubernetes Service): A managed service that simplifies running Kubernetes on AWS and on-premises, providing seamless integration with AWS services

Continuous Integration/Continuous Deployment (CI/CD) & GitOps

These tools automate steps in the software delivery process, such as initiating automatic builds, tests, and deployments to streamline the development lifecycle.

- Jenkins: A versatile open-source automation server, Jenkins facilitates building, testing, and deploying software, supporting a wide array of plugins for CI/CD purposes
- Argo CD: Specialises in Kubernetes-native workflows, enabling continuous delivery and automating deployment pipelines directly from Git repositories
- Argo Workflows: An extension of Argo as a workflow engine for orchestrating parallel jobs on Kubernetes
- Flux: Embraces GitOps principles by automatically applying changes from Git to Kubernetes, ensuring that the state of clusters matches the configuration stored in version control

Security and Compliance

Tools designed to enforce security and compliance within Kubernetes environments, from deployment to runtime.

- Falco: Detects abnormal application behaviour and potential security threats in real-time by monitoring system calls and Kubernetes audit logs
- Kubescape: Assesses Kubernetes clusters against known security standards and best practices, identifying compliance issues and vulnerabilities
- OPA (Open Policy Agent): A versatile policy engine that enforces policies across the entire stack, ensuring compliance with security policies and regulations

Storage Solutions

Storage solutions in Kubernetes are essential for data persistence and management in containerized environments, supporting dynamic provisioning and scaling.

- ETCD: Critical for Kubernetes, acting as the primary store for cluster state and metadata, ensuring high availability and consistency across a distributed environment
- Rook: An open-source cloud-native storage orchestrator for Kubernetes, providing the platform, framework, and support for a diverse set of storage solutions to natively integrate with cloud-native environments
- Ceph: A unified, distributed storage system designed for excellent performance, reliability, and scalability

Serverless and Event-Driven Architectures

These tools help in building applications that can scale from zero to planet scale without managing infrastructure, focusing on event-driven and serverless paradigms.

- KEDA (Kubernetes Event-Driven Autoscaling): Dynamically scales Kubernetes applications in response to events from various sources, optimising resource utilisation
- Knative: Enables the building and deployment of serverless and event-driven applications on Kubernetes, streamlining the development of cloud-native applications

Package Management and Application Deployment

Simplifying the deployment and management of applications in Kubernetes with tools that manage packages and dependencies.

- Helm: The Kubernetes package manager, Helm streamlines the deployment of applications through Helm charts, which define, install, and upgrade even the most complex Kubernetes applications