Skills Network

Glossary: OpenShift Basics

Term	Definition
A/B testing	Strategy is mostly used for testing new features in front-end applications. It is used to evaluate two versions of the application namely A and B, to assess which one performs better in a controlled environment. The two versions of the applications differ in terms of features and cater to different sets of users. Based on the interaction and responses received from the users such as feedback, you can choose one of the versions of the application that can be deployed globally into production.
Build	The process of transforming inputs into a resultant object.
BuildConfig	An OpenShift-specific object that defines the process for a build to follow. The build process makes use of the input sources and the build strategy. The BuildConfig is the blueprint, and the build is an instance of that blueprint.
Canary Deployments	Aims to deploy the new version of the application by gradually increasing the number of users. The canary deployment strategy uses the real users to test the new version of the application. As a result, bugs and issues can be detected and fixed before the new version of the application is deployed globally for all the users.
Circuit breaking	A method to prevent errors in one microservice from cascading to other microservices.
Configuration Change	A trigger that causes a new build to run when a new BuildConfig resource is created.
Control Plane	The control plane takes the desired configuration and its view of the services and dynamically programs and updates the proxy servers as the environment changes.
Custom build strategy	Requires you to define and create your own builder image.
Custom builder images	Are regular Docker images that contain the logic needed to transform the inputs into the expected output.
CRDs	Custom code that defines a resource to add to your Kubernetes API server without building a complete custom server.
Custom controllers	Reconcile the custom resources (CRDs) actual state with its desired state.
Data plane	Communication between services is handled by the data plane. If a service mesh is absent, the network cannot identify the type of traffic that flows, the source, and the destination and make any necessary decisions.
Enforceability (Control)	Istio provides control by enforcing policies across an entire fleet and ensures resources are fairly distributed among consumers.
Envoy proxy	All network traffic is subject to or intercepted by a proxy, called Envoy, used by the service mesh and allows many features depending on the configuration.
Human operators	Understand the systems they control. They know how to deploy services and how to recognize and fix problems.
Image Change	A trigger to rebuild a containerized application when a new or updated version of an image is available. For example, if an application is built using

	a Node.js base image, that image will be updated as security fixes are released and other updates occur.
ImageStream	An abstraction for referencing container images within OpenShift. Each image contains an ID, or digest, that identifies it. ImageStreams do not contain image data but rather are pointers to image digests.
ImageStream Tag	An identity to the pointer in an ImageStream that points to a certain image in a registry.
Istio	A platform-independent and popular service mesh platform, often used with Kubernetes. It intelligently controls the flow of traffic and API calls between services, conducts a range of tests and reduces the complexity of managing network services. Istio secures services through authentication, authorization, and encryption. Istio provides control by defining policies that can be enforced across an entire fleet. With Istio, you can observe traffic flow in your mesh so you can trace call flows, dependencies, and you can view service communication metrics such as latency, traffic, errors and saturation.
Man-in-the-middle attacks	A man-in-the-middle (MiTM) attack is a type of cyber-attack where the attacker secretly intercepts and relays messages between two parties who believe they are communicating directly with each other. The attack is a type of eavesdropping in which the attacker intercepts and then controls the entire conversation.
Observability	Helps to observe the traffic flow in your mesh, trace call flows and dependencies, and view metrics such as latency and errors.
OpenShift	A hybrid cloud, enterprise Kubernetes application.
OpenShift CI/CD process	Automatically merges new code changes to the repository, builds, tests, approves, and deploys a new version to different environments.
Operators	Automate cluster tasks and act as a custom controller to extend the Kubernetes API.
Operator Framework	Is a family of tools and capabilities to deliver an efficient customer experience. It is not just about writing code; what is also critical is testing, delivery, and updating Operators.
OperatorHub	Web console lets cluster administrators find Operators to install on their cluster. It provides many different types of Operators available, including Red Hat Operators, Certified Operators from independent service vendors partnered with Red Hat, Community Operators from the open-source community but not officially supported by Red Hat, and custom Operators defined by users.
Operator Lifecycle Manager	(or OLM) Controls the install, upgrade, and role-based access control (or RBAC) of Operators in a cluster.
Operator maturity model	Defines the phases of maturity for general day two Operations activities and ranges from Basic Install to Auto Pilot.
Operator Pattern	A system design that links a Controller to one or more custom resources.
Operator Registry	Stores CRDs, cluster service versions (CSVs), and Operator metadata for packages and channels. It runs in Kubernetes or OpenShift clusters to provide the Operator catalog data to OLM.
Operator SDK	(which includes Helm, Go, and Ansible) Helps authors build, test, and package their Operators without requiring knowledge of Kubernetes API complexities.
postCommit	Section defines an optional build hook.
Retries	A method to prevent errors in one microservice from cascading to other microservices.

runPolicy	Field controls how builds created from a build configuration need to run. Values include the default Serial (sequentially) and simultaneously.
Service Broker	Provides a short-running process that cannot perform the consecutive day's operations such as upgrades, failover, or scaling.
Service Mesh	A dedicated layer for making service-to-service communication secure and reliable. It provides traffic management to control the flow of traffic between services, security to encrypt traffic between services, and observability of service behavior; so, you can troubleshoot and optimize applications.
Software operators	Try to capture the knowledge of human operators and automate the same processes.
Source-to-Image	A tool for building reproducible container images. Also abbreviated S2i, it injects application source code into a container image to produce a ready-to-run image.
Source strategy	Section shows the strategy used to execute the build, such as a Source, Docker, or Custom strategy.
Source type	Determines the primary input like a Git repository, an inline Dockerfile, or binary payloads.
Webhook	A trigger that sends a request to an OpenShift Container Platform API endpoint. Often this will be a GitHub webhook, though it can also be a generic webhook. If a GitHub webhook is utilized, GitHub can send the request to OpenShift when there is a new commit on a certain branch, or a pull request is merged, or under many more circumstances. Webhooks are a great way to automate development flows so that builds can occur automatically as new code is developed.