GLOSSARY OF DEVOPS DEPLOYMENT STRATEGY TERMS

Unlock the Language of DevOps Deployment Strategies with our Comprehensive Glossary

Term	Definition
Blue/Green Deployment	A deployment strategy that involves creating two identical environments, one active (blue) and one inactive (green), and switching traffic from the old environment to the new environment by updating DNS or load balancer settings
Canary Release	A deployment strategy that involves releasing a new version of an application to a small subset of users, monitoring its performance and health, and gradually increasing the release to a larger audience
Rolling Deployment	A deployment strategy that involves updating a subset of instances at a time, verifying their performance and health, and continuing the update process until all instances are updated
Feature Flags	A deployment strategy that involves toggling certain features on and off for different users or groups, allowing for controlled release of new features and easy rollback if necessary
A/B Testing	A deployment strategy that involves releasing two versions of an application to different user groups, measuring their performance and user feedback, and choosing the version that performs better
Dark Launching	A deployment strategy that involves deploying new code to production but not making it publicly available, allowing for testing and monitoring before releasing it to users

Term	Definition
Zero Downtime Deployment	A deployment strategy that involves deploying new code or infrastructure without causing downtime or disruption to users
Immutable Infrastructure	A deployment strategy that involves creating and deploying infrastructure as pre-built images or templates, allowing for easier scaling, rollback, and versioning
Blue/Green/Canary	A combination of the Blue/Green and Canary release strategies, allowing for a gradual rollout of a new version to a small subset of users in the inactive environment (green) while maintaining the old version in the active environment (blue)
Rollback	The process of reverting to a previous version of an application or infrastructure, typically used as a fallback strategy in case of issues or failures during deployment
Red/Black Deployment	A deployment strategy that involves creating two identical environments, one active (red) and one inactive (black), and switching traffic from the old environment to the new environment by updating DNS or load balancer settings. This is similar to Blue/Green Deployment, but with the addition of verifying the new environment before routing traffic to it.
Continuous Deployment	A deployment strategy that automates the deployment process and releases new changes to production as soon as they pass the required testing and approval processes

Term	Definition
Infrastructure as Code	A deployment strategy that uses code to automate the deployment and management of infrastructure resources, allowing for versioning, collaboration, and consistency across environments
Hybrid Deployment	A deployment strategy that combines different deployment strategies, such as Blue/Green and Rolling, to optimize deployment speed and minimize downtime and disruption to users.
Immutable Deployment	A deployment strategy that involves creating and deploying pre-built images or templates of an entire application stack, including both the code and infrastructure, allowing for easy scaling, versioning, and rollback
Fast Fail	A deployment strategy that involves quickly detecting and stopping deployments that fail or cause issues, preventing further damage or disruption
Traffic Shifting	A deployment strategy that involves gradually shifting traffic from one environment or version to another, allowing for controlled release and testing
GitOps	A deployment strategy that uses Git as the source of truth for infrastructure and deployment configuration, allowing for versioning, collaboration, and auditability

Term	Definition
Chaos Engineering	A deployment strategy that involves intentionally introducing failures or disruptions to a system in a controlled environment, to test and improve its resilience and reliability
Release Train	A deployment strategy that involves synchronizing and releasing multiple applications or components as a single, coordinated release, to minimize disruption and ensure compatibility
Continuous Integration and Delivery (CI/CD)	A deployment strategy that involves continuously building, testing, and deploying code changes to production, typically through automation and collaboration tools and practices
Infrastructure Orchestration	A deployment strategy that uses tools and frameworks to automate the provisioning and management of infrastructure resources, allowing for scaling, monitoring, and management at scale
Serverless Deployment	A deployment strategy that involves deploying code to a serverless platform, such as AWS Lambda or Google Cloud Functions, without the need for managing servers or infrastructure
Release Pipeline	A deployment strategy that defines the stages and processes involved in deploying an application or component to production, typically including code review, testing, staging, and production

Term	Definition
Compliance as Code	A deployment strategy that uses code to automate the compliance and security policies and checks for an application or infrastructure, allowing for auditability, consistency, and scalability
Progressive Delivery	A deployment strategy that involves gradually rolling out new features or updates to users, using techniques such as canary releases, feature flags, or A/B testing, to reduce risk and ensure a smooth transition
Immutable Configuration	A deployment strategy that involves creating and deploying pre-built images or templates of application configuration, allowing for easy scaling, versioning, and rollback
GitOps Operations	A deployment strategy that extends GitOps to the entire operational lifecycle of an application, including monitoring, scaling, and remediation, to ensure consistency, visibility, and resilience
Shift-Left Testing	A deployment strategy that involves moving testing earlier in the software development lifecycle, to catch issues and defects earlier and reduce the risk of bugs or failures in production
Infrastructure Automation	A deployment strategy that uses tools and frameworks to automate the deployment and management of infrastructure resources, allowing for consistency, repeatability, and scalability

Term	Definition
Immutable Data	A deployment strategy that involves creating and deploying pre-built images or templates of data, allowing for easy replication, backup, and recovery
Blue/Green/Canary Analysis	A deployment strategy that combines Blue/Green and Canary releases with analysis tools, such as machine learning or statistical models, to automatically detect issues or anomalies and prevent them from affecting users
Hybrid Cloud Deployment	A deployment strategy that involves deploying an application or component across multiple cloud providers or on-premises environments, to leverage the strengths of each platform and ensure availability and resilience
Dark Launch Experimentation	A deployment strategy that extends Dark Launching to include experimentation tools, such as A/B testing or user feedback, to validate the new code or feature before releasing it to all users
Application Release Automation (ARA)	A deployment strategy that automates the entire application release process, from code check-in to production deployment, to reduce errors, increase efficiency, and ensure compliance
Self-Healing Infrastructure	A deployment strategy that uses automation and monitoring tools to detect and automatically fix issues or failures in the infrastructure or application, reducing downtime and ensuring resilience

Enjoyed this?

One favor to ask...

Sharing
=
caring

"Be a good friend.

Support free content with a repost."

in @devops-expert