

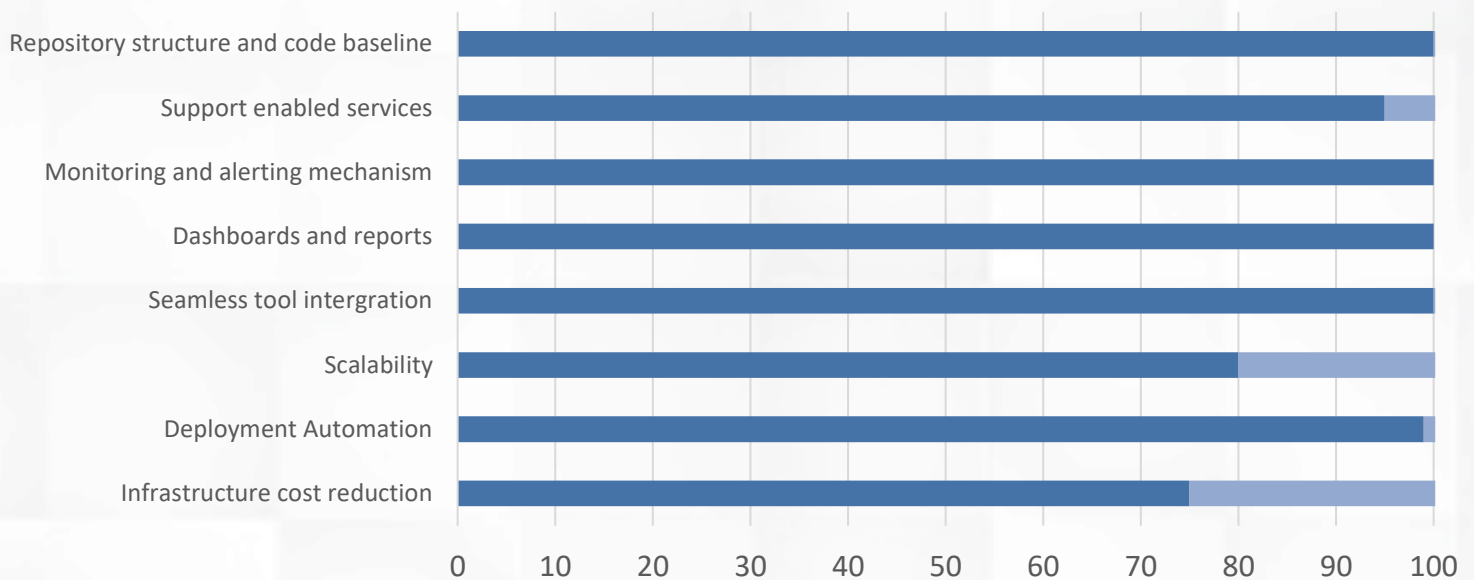


MISSION & VISION

DLITES is an end-to-end framework with a suite of tools and services which enables customers to establish a robust and concrete modernized DevOps, SRE practice using Microsoft Azure services.

DLITES handles all the aspects of DevOps, Live Site Engineering and Support system which enables customers to have a smooth deployment and monitoring processes.

DLITES Features



FEATURES

Migrating existing classic pipelines to YAML pipelines:

Flexibility across platforms/projects, version control for pipelines, Single pipeline model for CI/CD, Reusability, multistage pipeline.

Dynamic pipelines instead of static pipelines:

Avoiding hardcoded values, accessing environment related configurations, Application specific configurations etc.. via variables and variable groups in centralized location.

Implementing Security code scan:

Helps in detecting various security vulnerability patterns, SQL injection, Cross-Site Scripting etc. in application.

Implementing Branching & Security policies:

Enforcing proper branching and security policies for users.

Linking Azure Key vaults along with variable groups:

Accessing secrets dynamically related to any sensitive information in pipelines in more secure & standard manner.

Implementing Quality gates using Sonar Cloud:

Measuring & analyzing the source code quality, reduces the risk of software development.

Infrastructure provisioning (Iac): Creating pipeline for setting up and configuring infrastructure for different environments with minimal manual intervention using ARM template , terraform etc.

Addressing Security issues from Software Composition Analysis (SCA): Identifying the security concerns from packages in code like related artifacts, registry, licenses, compliance data using tools like OWASP & others.

Validating code changes using Deployment slots :
Validate webapp changes in staging deployment slot before swapping it with production slot.

Automating custom test cases through Azure test plan:
Automating required custom test cases to validate the application after deployment or release process.

IMPLEMENTATION PROCESS

14-18 Days

- Assess and evaluate the repository and code base.
- Prioritize scope and build a plan that addresses value, impact, and risk.
- Prototype build, deploy, and migrate work patterns.









60 Days

- Develop build and deploy setup.
- Complete automation of deployments using scripts.
- Continuous Integration and Continuous Deployment model.

120-180 Days

- End to End automation using tools and scripts.
- Enablement of dashboards and monitoring tools.
- Setup a SRE process to provide continuous monitoring and release support.

PRINCIPLES AND PRACTICES

Continuous planning 	Continuous integration 	Continuous delivery 	Continuous operations 			
<ul style="list-style-type: none"> • Objectives & Key Results (OKR) • Lean product discovery • Lean product definition • Release planning • Sprint planning • Agile requirements • Security requirements • Architecture design • Capacity planning • UX architecture design • Threat modeling • Prioritization & estimation • Demos & Retrospectives 	<ul style="list-style-type: none"> • Behavior-driven development (BDD) • Test-driven development (TDD) • Microservices & container development • Mono-repo & Multi-repo • Unit testing & code coverage • Version control • Git pull request • Trunk-based policies • Security static code scan • CredScan • Open Source software (OSS) component compliance • Build parallel & serial pipeline 	<ul style="list-style-type: none"> • Release pipeline • Secure infrastructure deployment • IaaS deployment • PaaS deployment • Shared services • Infrastructure as code (IaC) • Change management • Configuration management • Release management • Blue-green deployments • Canary deployments • Feature flags • Trunk production ready 	<ul style="list-style-type: none"> • Site reliability engineering (SRE) • Telemetry & monitoring • Application performance monitoring • Auto Failover, scaling, & DR • Modern service management • Secure access & application data • High availability, security, cost & performance advisory • Secure DevOps ChatOps • Shift-right testing • Secrets management • Governance & GDPR support • Automation & AIOps • Continuity & resilience 			
Continuous quality 	Quality requirements	Shift-left testing	Governance & standards	Test automation	Compliance & audits	Shift right testing
Continuous security 	Security architecture	Access & identity management	Application & data security	Security infrastructure	Secure operations	Governance, risk, & compliance
Continuous collaboration 	Collaborative culture	Alignment & autonomy	Kanban collaboration	Wiki & Teams collaboration	ChatOps collaboration	Feature Team & SRE
Continuous improvement 	Lead time & cycle time	Deployment frequency	Mean time to restore (MTTR)	Change fail percentage	Continuous feedback	Value stream mapping