Structures that would utilise 3rd party DevOps tool:

1. CI/CD
2. Source Control
3. Configuration Management
4. Containerisation
5. Code Quality and Testing
6. Monitoring and Logging
7. Security and Compliance
8. Infra Automation and Management

If you use Jenkins for CI/CD,

for Source Control, GitHub or GitLab can be used

For config- if terraform is already integrated w jenkins, then its overall additional plugins and toolkit complexity and overhead can be ignored
Terraform can prove ineffective while dealing with collaborative distributed systems, thus SaltStack is better suited for a centralised control over multiple collaborative systems. Integration with Jenkins by plugins and or invoking Salt commands as build steps

For Containerisation: Docker is portable , shares the host OS's kernel therefore better resource utilisation compared to VMs
Kubernetes has self-healing mechs that automatically replace and restart failed containers. High availability via distribution of workloads across multiple nodes
Better for advanced scalability and reliability (more setup and management overhead)
can also use EKS, advantage: managed upgrades and patching

> If you use Kubernetes then for infra-automation and management Kops can be used<- easier deployment of clusters, seamless integration of IAM for access control (mostly all of the AWS ecosystem can be leveraged). Also supports horizontal scaling dynamically based on workload demands.

For code quality and testing: for pure code quality analysis: SonarQube keeps good check of code health, linters for various languages can be used too for the same. SonarQube also scans the code for any vulnerabilities and compliance issues in the code.

> For web app testing: Selenium is highly recommended

For monitoring and logging: Grafana+Prometheus would provide visual analytics along with dashboarding and monitoring. Only if configured with external plugins supporting offline alerting such as SMS gateways, their reliance on internet can be removed for alerting (unexplored)

> PagerDuty is a better alternative here given its central strength being incident management, escalation and alerting. Can and does alert without internet access

The ELK stack has to have another third-party integration in order to send alerts, no native offline alerting capabilities

For Security and Compliance: Veracode integrates with CI/CD pipelines to enable DevSecOps easily, offers static, dynamic and software composition analysis for vulnerabilities. Provides role-based access control, API integration, centralised reports and analytics

> For container security: Twistlock- specific for containerised applications; runtime protection and anomaly detection. Synk focuses on open-source dependencies' vulnerabilities, ideal for orgs that require ongoing visibility

List of 3rd Party Tools:

github	maven	jenkins	chef	puppet	ansible
terraform	docker	k8s	slack	awsCloudFormation	

GCP	SignalFx	Splunk	Selenium	Gremlin	ELK
phantom	nagios	vagrant	gradle	prom	bamboo
jira	pulumi	SaltStack	CircleCI	Loki	PagerDuty
Scalr	env0	gitpod	synk	doppler	flux
broadcomRally	TeamCity	TestSigma	awsCodeDeploy		

Another Company's DevOps Architecture

DevOps tool: Jenkins, GitHub, AWS Code Deploy, Terraform & Python for Code Scripting
Instance types depend on requirements, we mostly use T3 instance types
Alarming: use CloudWatch service along with SNS topic

I looked into different blogs posted by quite a few of the MAANG and allied companies. They use custom deployment scripts or another tool for CD part of the CI/CD: Jenkins for CI still (BuildKite was also suggested)
Since end user functionality of each app from Netflix to Uber is different, I did not make a note of the instance types

Jaeger is used by Uber for tracking distributed tracing amongst their architecture