

Santosh Katageri

Senior DevOps / Platform Engineer

santoshpkatageri@gmail.com | [LinkedIn](#) | [GitHub](#) | [learnwithsk.dev](#)
Hubballi, Karnataka, India | +91 88806 08429

SUMMARY

Senior DevOps/Platform Engineer with 9+ years building and operating cloud platforms across OCI, AWS, and GCP. At Oracle, I own the deployment automation layer for two major services — **OCI Data Catalog**, a public data-governance service bootstrapped across ~100 regions, and the new **OCI AI Data Platform**, taken from a monolithic bootstrap to isolated deployment units. Author of **two published Terraform providers** in Oracle's public provider repo (2021–present); hands-on with Prometheus/Grafana observability. I build developer-facing tooling — CLIs, self-heal automation, MCP workflows — and publish the builds at [learnwithsk.dev](#). M.Tech (Software Engineering), BITS Pilani.

EXPERIENCE

Senior Member of Technical Staff, Platform & Cloud DevOps Oracle Corporation

Jan 2020 – Present ■ Bengaluru

Promoted Jan 2022; joined as Member of Technical Staff — builds and operates OCI Data Catalog and OCI AI Data Platform.

- Drove OCI Data Catalog's regional bootstrap from semi-manual (~3 days/region) to **touchless automation** — authored the service's Terraform modules for Oracle's internal IaC platform and the touchless execution flow; ~100 OCI regions bootstrap in **120 minutes**, with recent builds under 90 minutes against the 120-minute SLA.
- Cut a critical bootstrap stage from 50 to 20 minutes (**–60%**) via burstable capacity, parallel (non-sequential) execution for new regions; built **shape-agnostic autoscaled orchestration** bootstrapping computes on dynamic availability across VM architectures.
- Migrated bastion access from jump-hosts to **Zero Trust Bastion** and made bootstrap configuration realm-agnostic across 60+ realms (co-located datacenter, dedicated, government, university, single-rack); built a **Python region validator** and release-validation suite gating both touchless bootstrap and QA → preprod → prod release promotion during upgrades.
- Built an **ephemeral simulated environment** for end-to-end bootstrap validation — the full service stack created, destroyed, and re-created in isolation per rollout; authored isolation guidelines for non-deletable services to keep re-creation safe.
- Implemented **Prometheus/Grafana** service-health metrics and canary alarms, improving detection and triage of region-build failures and downstream dependency issues.
- Author of **two published Terraform providers** (OCI Data Catalog, AI Data Platform) in Oracle's public terraform-provider-oci — schema design, API/SDK compatibility, async work-request handling, resource locking, releases, and direct customer support; active contributor 2021–present.
- Decomposed OCI AI Data Platform's monolithic bootstrap into isolated, interdependent deployment units across a **20+ microservice** estate — multi-VCN networking, autoscaled control-plane compute, cell-based data-plane networking, ATP, Kubernetes, KMS/certificate management, DNS, IAM, and internal packaging/KV/workflow/limits services — deployed via **Helm** through OCI DevOps with region validation and canary deployments.

KEY ACHIEVEMENTS

01

Region-build acceleration

Drove OCI Data Catalog's bootstrap from ~3 days/region to **120 minutes** across ~100 regions — recent builds under **90 minutes**, beating the SLA.

02

Platform decomposition

Took the AI Data Platform's monolithic bootstrap (**20+ microservices**) to isolated deployment units — **90% automated**, realm-agnostic across 60+ regions and 12+ realms.

03

Two published Terraform providers

Authored providers for Data Catalog and AI Data Platform in Oracle's public terraform-provider-oci; active contributor 2021–present.

04

36–50% cloud cost cuts

Automated AWS waste cleanup at Zynga (**–36%**) and EC2 scheduling automation at Mindtree (**–50%**).

SKILLS

Cloud:	OCI	AWS	GCP
IaC:	Terraform	Provider development	Config mgmt
CI/CD:	OCI DevOps	GitHub Actions	Jenkins
	Helm	Canary deployments	Release management
Observability:	Alarm design & deployment	Prometheus	
	Grafana	On-call & incidents	Service-Health Reviews
Containers & OS:	Linux	Docker	Kubernetes
Languages:	Python	Go	Bash/Shell
			Cloud CLIs/SDK
Networking & Edge:	Cell-based Multi VCNs	DNS	
	Akamai/Cloudflare edge	Zero Trust Bastion(ZTB)	
Security:	IAM & policy automation	Zero Trust access	
	KMS vault & secrets management	SSH certificates	
Tooling:	Git	Jira	Confluence
		LaTeX	
	AI-assisted engg (Codex, MCP)		

RECOGNITION

Outstanding Performer of the Year

Mindtree

Spot Award — Major Security Incident Resolution

Zynga

DevOps Hackathon Winner

TechGig

- Lifted AIDP onboarding from 1–2 days of manual setup to **90% automated** — retired most manual blockers en route to full touchless; realm-agnostic deployment across **60+ regions and 12+ realms**.
- Led a multi-team rollout of the Punchout UI for the AI Data Platform data plane — OCI console origin service, **Akamai CDN**, internal certificate service, and regionless DNS — fully automated through OCI DevOps.
- Built developer tooling: an on-call assistant (service monitoring + scheduled Jira polling) and a local-first CLI converting runbooks into guarded, Git-reviewed, recoverable workflows with explicit validation checkpoints.
- Partnered with security and compliance on region rollout compliance; contribute to release upgrades, monthly security patching, hotfixes, and production on-call; mentor and onboard engineers on deployment automation and troubleshooting; host weekly service-health reviews.

Cloud Engineer

Zynga Inc.

Nov 2018 – Jan 2020 ▪ Bengaluru

Promoted Jun 2019; joined as System Engineer.

- Designed and built automated cleanup of underutilized AWS resources, cutting cloud infrastructure spend by **36%**.
- Designed and implemented an IAM automation system using AWS Lambda, Terraform, and Jenkins.
- Developed cloud automation in Python (Boto3) for multi-cloud infrastructure supporting high-traffic game teams across AWS and GCP.
- Earned a **Spot Award** for resolving a major security incident.

DevOps Engineer

Mindtree Ltd.

Oct 2016 – Nov 2018 ▪ Bengaluru

Promoted Jun 2018; joined as Junior DevOps Engineer.

- Built EC2 start/stop scheduling automation that reduced a client's cloud infrastructure costs by **50%**.
- Delivered automation (Shell, Python, Jenkins, Ansible) as part of the DevOps Center of Excellence; supported CI/CD setup and release pipelines across dev/test/prod for transportation and car-rental clients.

PROJECTS

LearnWithSK.dev

live · learnwithsk.dev

An engineering publication on building and understanding real systems — **articles**, **working notes**, and **build logs** from live projects. Static site on **Cloudflare Pages** with per-branch preview deployments and GitHub Actions CI/CD.

Ship Boring

in development · [github](https://github.com)

Building an automated QA → preprod → prod promotion platform on EKS — Terraform, ArgoCD GitOps, Kyverno gates, **metric-gated canaries with automatic rollback** (Argo Rollouts). Build-in-public series **Build, Break & Operate**: learnwithsk.dev.

DevPersonify

live · devpersonify.learnwithsk.dev

GitHub history → portfolio audit + job-ready documents, **deterministic scoring**, **no AI black box**, **no backend** — shipped end-to-end via AI-assisted development.

EDUCATION

M.Tech, Software Engineering

BITS Pilani

Jul 2022 – Jun 2024

Work Integrated Learning Programmes (WILP)

B.E., Telecommunication Engineering

Siddaganga Institute of Technology, Tumkur

2012 – 2016

FOCUS AREAS

IA

Infrastructure automation

Touchless provisioning, shape-agnostic orchestration, ephemeral validation environments — automation as the default, manual as the exception.

DA

Deployment architecture

Designing bootstrap and promotion systems: environment topology, isolation boundaries, gated rollouts, recovery paths.

DX

Developer platforms

The tooling layer — CLIs, validation gates, runbook tooling — that makes multi-region delivery self-service and safe.

WR

Writing in the open

Publishing build logs and working notes at learnwithsk.dev — complex systems, explained plainly.

REFERENCES

Available on request — recommendations visible on my [LinkedIn profile](#).