OCI Fleet Patching Automation

Design and automate multi-region fleet patching workflows on Oracle Cloud Infrastructure to minimize manual effort, accelerate deployment cycles, and enhance operational reliability across distributed environments.

View Implementation

See Results

Core Goals

Automated Orchestration

Develop fully automated patch orchestration using Terraform IaC and Python SDK for seamless deployment.

Standardization

Standardize patch configuration, update validation, and deployment consistency across multiple OCI regions.

Containerization

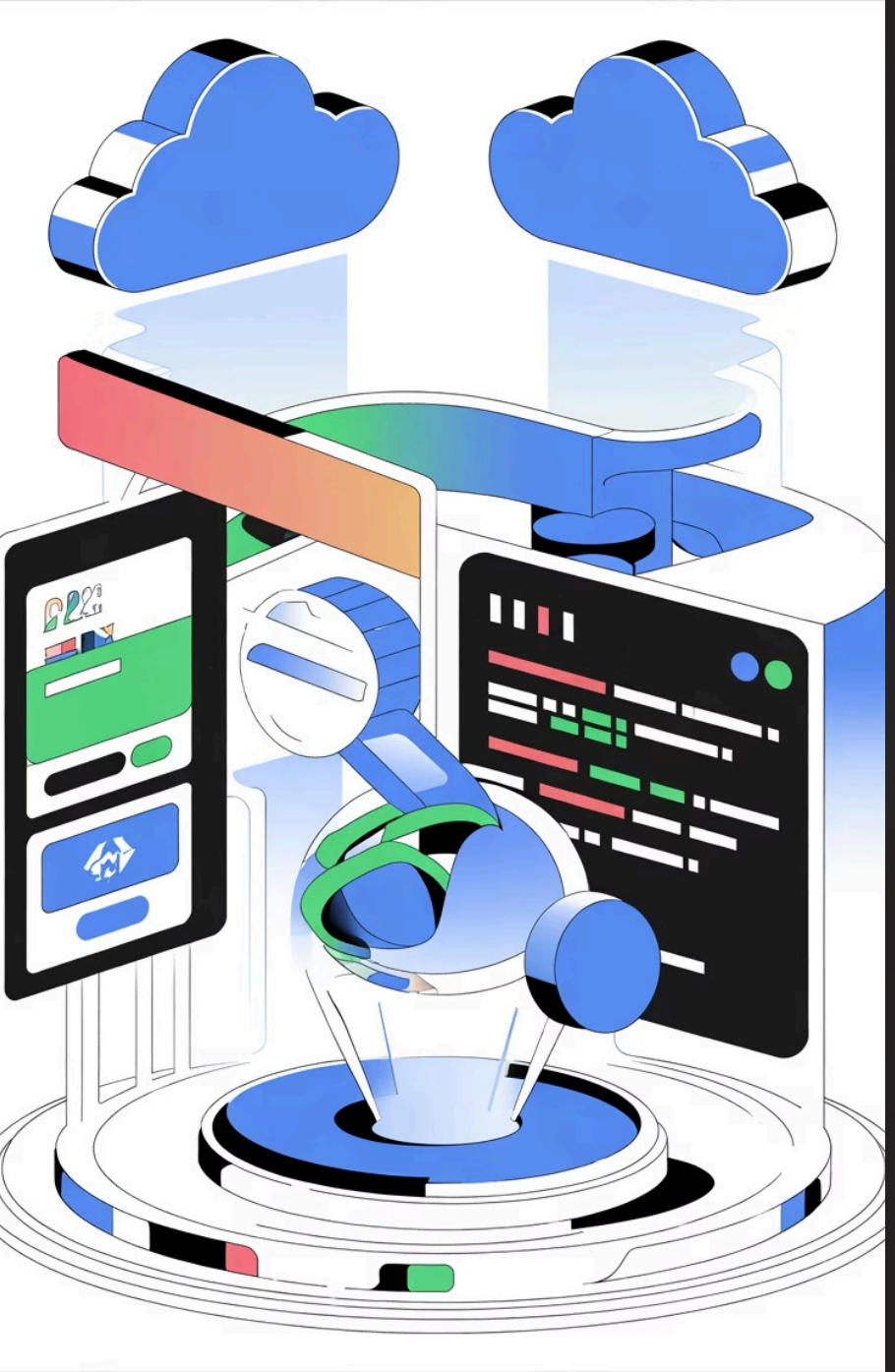
Package automation tools and patching scripts within Docker containers for consistent runtime environments.

Orchestration

Utilize Kubernetes orchestration to manage distributed patch workflows and parallel job execution.

Telemetry Monitoring

Implement OCI Telemetry Alarms for real-time health monitoring, alerting, and performance visibility.



Technology Stack

Cloud & Infrastructure

Oracle Cloud Infrastructure, Terraform, OCI Telemetry

Development & Scripting

Python SDK, Bash, YAML

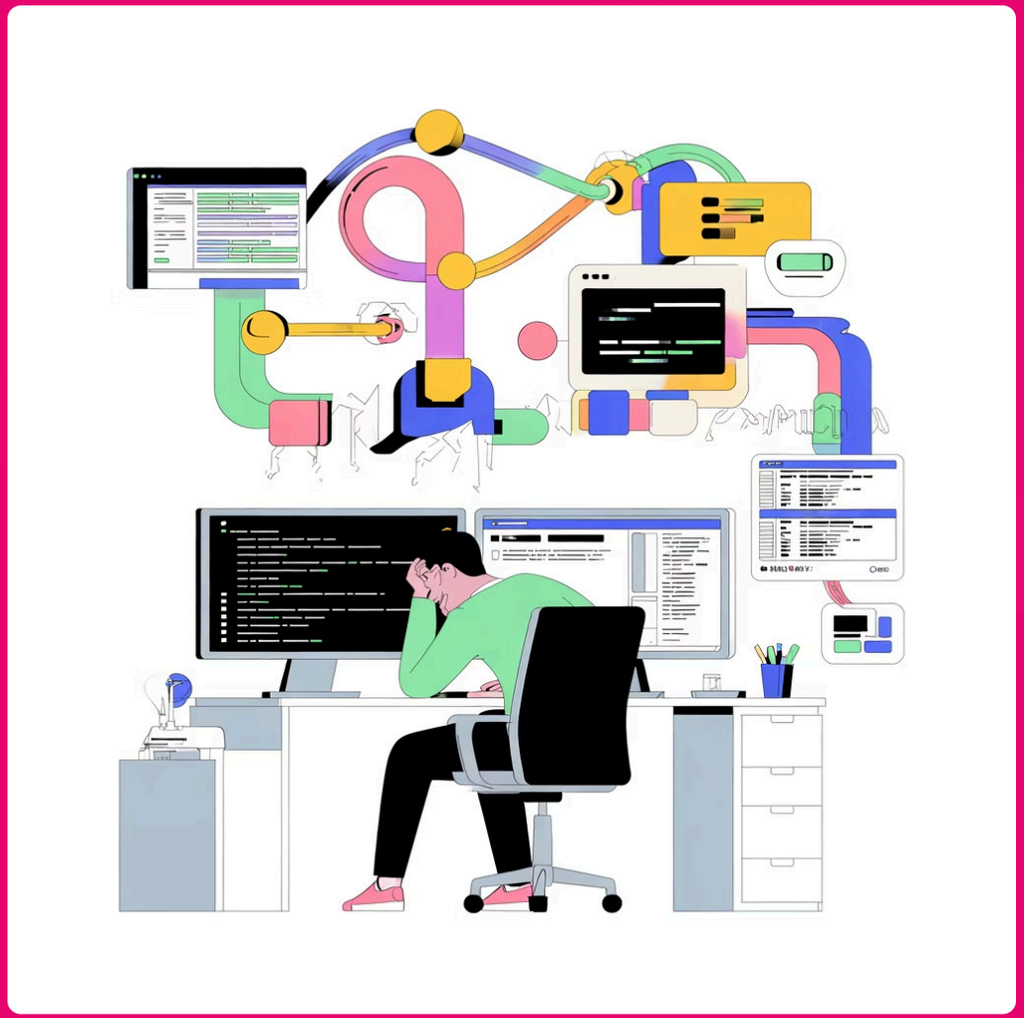
Containerization & Orchestration

Docker, Kubernetes

CI/CD & Version Control

GitHub Actions

Challenges Addressed



Manual Processes

Manual, region-specific patching processes causing delays and inconsistencies across deployments.

Limited Observability

Limited observability and alerting for failed patch executions impacting response times.

Environment Inconsistency

Lack of containerized environments for reproducible automation across regions.

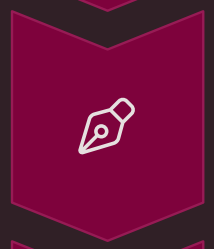
Scalability Gaps

Need for self-healing automation and cross-region scalability to support growth.

Architecture Flow

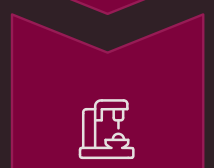
The automation pipeline orchestrates patch deployment across multiple regions with integrated monitoring and self-healing capabilities.

Python SDK Trigger

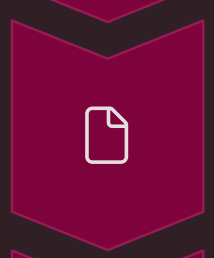


Terraform IaC Deployment

OCI Compute Fleet



Dockerized Pipeline



Kubernetes Orchestration

OCI Telemetry Alerts

Implementation Highlights

01

Terraform & Python Automation

Developed Terraform modules and Python SDK scripts to automate OCI Fleet Patching workflows across multiple regions.

02

Docker Containerization

Packaged patch management logic into Docker containers, ensuring environment parity during deployments.

03

Kubernetes Orchestration

Built Kubernetes orchestration pipelines to execute parallel patch jobs, handle node failures, and perform auto-retries.

04

Telemetry Integration

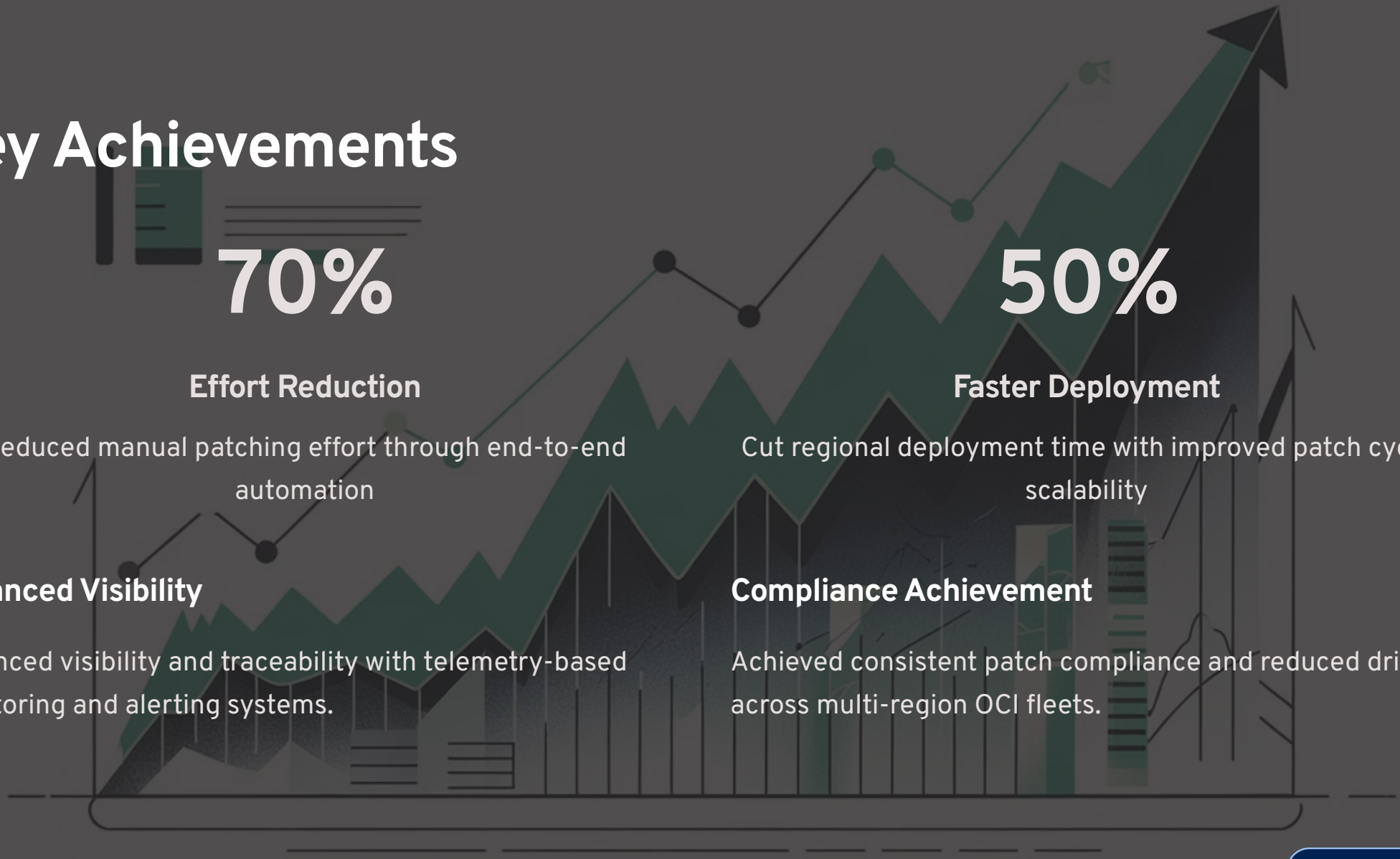
Integrated OCI Telemetry Alarms and logs for continuous system health and SLA tracking.

05

CI/CD Workflows

Enabled GitHub Actions CI/CD workflows for version-controlled deployments and automated validation.

Key Achievements



70%

Effort Reduction

Reduced manual patching effort through end-to-end automation

50%

Faster Deployment

Cut regional deployment time with improved patch cycle scalability

Enhanced Visibility

Enhanced visibility and traceability with telemetry-based monitoring and alerting systems.

Compliance Achievement

Achieved consistent patch compliance and reduced drift across multi-region OCI fleets.

Outcome & Impact

Established a **reliable, scalable, and self-healing OCI Fleet Patching Automation framework** that aligns with Oracle's reliability and compliance goals.

This initiative paved the way for **autonomous patching** and cross-cloud DevOps integration, setting new standards for operational excellence.

Skills & Tools Showcased

Cloud Infrastructure

Oracle Cloud Infrastructure (OCI), Multi-Region Scalability

Infrastructure as Code

Terraform, Python SDK, YAML

Containerization

Docker, Kubernetes

DevOps & CI/CD

GitHub Actions, Automation

Observability

OCI Telemetry, SRE

Ready to Transform Your Infrastructure?

This OCI Fleet Patching Automation project demonstrates expertise in cloud infrastructure, DevOps practices, and enterprise-scale automation. The framework delivers measurable improvements in efficiency, reliability, and operational excellence.

Get in Touch

Learn More