



Accelerating Blockchain PoC Development with Infrastructure Automation

Automating infrastructure provisioning, Kubernetes setup, and CI/CD pipelines to rapidly and consistently deliver blockchain Proof-of-Concepts (PoCs), accelerating innovation cycles.

Overview

A global private blockchain technology provider needed to dramatically accelerate the setup of their Proof-of-Concept (PoC) environments to quickly demonstrate use cases for diverse enterprise clients.

- Designed a fully automated, configuration-driven solution that streamlines the creation of PoC environments using a simple JSON configuration file and Python automation scripts.
- Integrated robust automation tools including Terraform for infrastructure, Ansible for Kubernetes setup, and Jenkins for CI/CD pipeline orchestration.
- Reduced PoC setup time from several days to just a few hours, significantly increasing efficiency and accelerating innovation cycles.
- Improved consistency, eliminated repetitive DevOps overhead, and enabled rapid, scalable blockchain experimentation.



Client Profile

A leading private blockchain software technology provider with a global footprint.

Challenges: Manual Setup Slowed Innovation

The client's manual approach to environment provisioning created significant bottlenecks.

- **Slow Time-to-Market:** Each PoC required days of manual, repetitive work for DevOps teams to set up Git repos, K8s clusters, and pipelines.
- **Operational Inconsistencies:** The lack of a standardized, automated approach led to configuration drift and inconsistencies across different client environments, increasing debugging time.
- **High DevOps Overhead:** The manual burden on the DevOps team diverted resources from strategic tasks to repetitive environment provisioning.

- **Hindered Agility:** The slow delivery process hindered the client's ability to quickly explore and demonstrate the potential of new blockchain use cases.

Solution: Automated Blockchain PoC Framework

We designed and implemented a comprehensive, automated provisioning framework leveraging Infrastructure as Code (IaC) and configuration management to accelerate PoC delivery and ensure consistency.

The entire process is driven by a simple, centralized JSON configuration file and executed by a core Python automation script.

- **Configuration-Driven Setup:** A developer simply updates the JSON file, and the Python script drives the entire end-to-end automation process.
- **Infrastructure Provisioning:** Terraform was used to reliably provision the necessary Azure Virtual Machines (VMs) for the PoC environments.
- **Kubernetes Cluster Setup:** Ansible was used for custom configuration and setup of Kubernetes clusters in environments where Azure Kubernetes Service (AKS) was not available.
- **CI/CD Integration:** Jenkins pipelines were automatically created, linking to automated Git repository creation and leveraging Helm charts to deploy the Docker images into the Kubernetes clusters.

Technical Highlights

The solution was built for speed, consistency, and extensibility.

- **Automation Stack:** Python (orchestration script), Terraform (IaC for VMs), and Ansible (Configuration Management for K8s).
- **CI/CD Pipeline:** Automated creation of Jenkins jobs, Dockerfiles, and Helm Deployments for consistent build and release cycles.
- **Configuration Flexibility:** The JSON-driven approach ensures that new PoC environments can be rapidly spun up consistently and securely by simply changing input variables.

- **Containerization:** Utilized Kubernetes (K8s) for scalable, isolated application hosting.

Impact

The automated framework dramatically streamlined the client's innovation process.

- **Accelerated Delivery:** PoC setup time was reduced from several days to just a few hours, enabling much faster response to client requests.
- **Increased Efficiency:** Eliminating repetitive manual tasks significantly reduced DevOps overhead, allowing the team to focus on higher-value activities.
- **Improved Consistency:** The IaC/GitOps approach ensured uniformity across all PoC environments, minimizing errors and speeding up debugging.
- **Faster Innovation Cycles:** The ability to rapidly spin up environments enabled the client to explore more blockchain use cases and demonstrate potential quickly.