Denis Speranskiy

Senior Infrastructure Engineer & Go Developer

speranskiy@gmx.com
+(382) 68 505 821

SUMMARY

- With over a decade of experience in the IT industry, my greatest strength is tackling complex problems head-on. My career evolved from hands-on Deployment Engineering to Team Lead at HeadPoint LLC - managing infrastructure for a city-wide video surveillance platform and an enterprise IoT platform serving banking, retail, and municipalities. Since April 2023, I have worked as a Senior Infrastructure Engineer at Aristocrat (via Itransition), designing and delivering solutions for a heavily loaded, GCP-based platform. I collaborate closely with Dev, QA, and project managers to ship changes end-to-end, and regularly investigate complex incidents spanning infrastructure and application tiers using the LGTM observability stack. More recently, I have been channeling this mindset into backend software development with Go - building production-grade services at the intersection of infrastructure and application engineering.
- Languages: English, Russian

SKILLS

Infrastructure
GCP, AWS, VMWare, Yandex Cloud, Huawei, Mikrotik, Fortigate

Orchestrators
Kubernetes, Docker Swarm

Tools
Packer, Terraform, Ansible, Kustomize, Helm, Crossplane, KEDA, GitHub Actions, Argo Workflows

Languages
Go, Python, Shell, JavaScript, Groovy

Apps
Krakend, Traefik, GlusterFS, NFS, CIFS, S3, MongoDB, PostgreSQL, Redis, Kafka, RabbitMQ, HashiCorp Vault

WORK EXPERIENCE

Senior Infrastructure Engineer — Apr 2023 - Present

Itransition

Aristocrat

- A leading gaming content creation company powered by technology to deliver industry-leading casino games together with mobile games and online real money games.
- Geo-distributed application deployment across multiple GKE clusters utilizing a geo-sharded MongoDB architecture, managed via ArgoCD with advanced application set configurations. API exposed through KrakenD with JWK authentication. Managed zero-downtime Kubernetes cluster version upgrades and automated disaster recovery switchovers across production environments. (GCP, Helm, Crossplane, Kubernetes, Mongodb, ArgoCD)

- Migrated a legacy Java application to Kubernetes using GitOps. The application consists of a frontend, workers, services, cron jobs, and batch jobs. A GitHub Action was implemented to version and deploy the frontend as a single-page application to a storage bucket. Worker scaling is automated based on RabbitMQ queue metrics to ensure optimal resource utilization. (GCP, Helm, ArgoCD, Kubernetes, Java, RabbitMQ, Mongodb)
- Enhanced the CI/CD platform by integrating Argo Workflows as a cluster add-on with RBAC-based access control, fitting seamlessly into the in-house deployment framework to enable efficient rolling updates for both infrastructure and external teams. (ArgoCD, Helm, RBAC, GitOps)
- Migrated Terraform IaC to Crossplane for declarative resource management, including automated BigID datasource provisioning. Migrated from Kubernetes Ingress to Gateway API with Service Mesh enabled across all environments, with load testing to validate zero-downtime routing and traffic mirroring for safe production rollouts. Certificate management integrated as part of the Gateway API solution. (Crossplane, GCP, Terraform, Kubernetes)
- Developed and restructured internal Developer Portals using Templ+HTMX to empower QA and engineering teams with self-service environment provisioning and direct Kubernetes API integrations. (Go, HTMX, Kubernetes, GCP)
- Designed and implemented autoscaling architectures utilizing KEDA and GCP Memorystore autoscalers to dynamically handle high-volume traffic spikes based on queue depth and memory metrics. (Kubernetes, KEDA, GCP, RabbitMQ, Kafka)
- Led database architecture modernizations: decoupled monolithic databases to dedicated MongoDB clusters and upgraded legacy SQLite engines to MySQL for improved I/O performance. Automated Atlas MongoDB log collection using Cloud Run Functions, Pub/Sub, and Cloud Scheduler. (MongoDB, MySQL, GCP, Terraform)
- Standardized secret and certificate management via HashiCorp Vault and GCP KMS; configured fleet-wide rsyslog routing on GCE VMs for enterprise audit compliance. Drove cost optimization through proactive infrastructure audits to decommission idle resources and right-size workload replicas. (HashiCorp Vault, GCP, Kubernetes, Terraform)
- Built an internal deployment tool in Go integrating with the GitHub API to trigger CI/CD pipelines deploying services to Kubernetes clusters and GCS buckets. Exposes a REST API consumed by external production tooling, surfaces live service state via the Kubernetes API, and enforces role-based access via OKTA. UI built with HTMX and Go templating. (Go, Kubernetes, GCP, GitHub Actions, HTMX, OKTA)

Viclarity

- Infrastructure management for a provider of Governance, Risk, and Compliance (GRC) software solutions. Responsibilities included optimizing pipelines, AWS configurations, infrastructure-as-code and cost efficiency. Led the planning and migration strategy from EC2 to EKS. Managed CI/CD tool upgrades and orchestrated the migration of services from one server to another. (AWS, EC2, CloudFront, CloudFlare, CodeDeploy, Teamcity, Terraform, Packer)

Team Lead Jul 2018 - Mar 2023

HeadPoint, LLC

IoT and Video products

- Developed a universal deployment platform from the ground up, serving both as a source for development and test environments and for production use on customer sites. The platform abstracts the underlying infrastructure and deployment processes for various service types (Docker, Windows Services, systemd units), providing development, QA, and business teams with an intuitive UI to deploy and test services without worrying about the underlying deployment specifics or environment. (Jenkins, Artifactory, Terraform, Ansible, Gitea, Python, Packer)
- Linux and Windows server fleet management and support using Ansible across approximately 200 VMs, covering configuration drift remediation, patch management, and service reliability. (Ansible, NFS, DNS, Traefik)
- Migrated full infrastructure configuration to Infrastructure as Code, replacing ad-hoc manual provisioning with reproducible, version-controlled definitions across all environments. (Terraform, Packer, VMWare, Ansible, Docker Swarm, Kubernetes, GitOps)
- Version and release control system utilizing automatic semantic versioning based on conventional commits, automated changelog generation, and notifications for new app versions. The system provides transparency in how environments are updated and managed, eliminating the need for developers to maintain dedicated Git branches to keep environments synchronized. (Git, Python, Confluence, Jenkins, Slack, Mattermost)
- Migrated to a self-hosted Kubernetes cluster using the GitOps methodology. To leverage Kubernetes benefits within our closed infrastructure, I deployed a Talos-based Kubernetes cluster on VMWare and developed an LDAP integration service for seamless team authorization. (Kubernetes, VMWare, GitOps, ArgoCD, Kustomize, Helm, Talos, Go)
- Designed and implemented a public-facing multi-tenancy installation of our product on YandexCloud, leveraging managed solutions for cost optimization. The primary challenge was supporting multiple VPN tunnels and network configurations to enable secure access for clients from the cloud to their own infrastructure. (YandexCloud, Terraform, Packer, Networking, S3)

TRAININGS AND CERTIFICATIONS

Certified Kubernetes Administrator 2023

https://www.credly.com/badges/320dcaec-6541-4239-8dbe-4d619db17da3/public_url

Earners of this designation demonstrated the skills, knowledge and competencies to perform the responsibilities of a Kubernetes Administrator. Earners demonstrated proficiency in Application Lifecycle Management, Installation, Configuration & Validation, Core Concepts, Networking, Scheduling, Security, Cluster Maintenance, Logging / Monitoring, Storage, and Troubleshooting (Kubernetes)

SIDE PROJECTS

Receipt Processing Application 2025

https://receipts.speran.info

Personal receipt-processing application built in Go following production best practices: structured as discrete components (migration tool, background worker, API service) communicating via Redis. Integrates with Telegram and a web UI for receipt ingestion, applies LLM-based categorization, and persists data in a relational database with full authentication, per-user token management, and multi-user federation (family) support. UI built with HTMX and Go templating, minimizing JavaScript surface area while retaining

Ubuntu Team Russia 2012

https://ubuntu.ru

Volunteered as a Forum Administrator at forum.ubuntu.ru. Provided guidance and assistance to users in their journey with the operating system. Contributed to fostering a collaborative and supportive environment within the community (Linux, Ubuntu, Docs)