ventx GmbH

@ Munich ReThink GmbH

Project and Consulting DevOps/Cloud

ventx
we make IT

Company data

Established: 2014

IT Service Provider for DevOps & Cloud

AWS Consulting Partner

ventx
we make IT

Infrastructure

- **I**nfrastructure **a**s **C**ode (*"IaC"*):
  - Terraform, Ansible, CloudFormation
  - Immutable, reproducible, consistent infrastructure
- Self-healing infrastructure
- Multi-Tenant, Multi-Stage Terraform stack
- Multi-Cloud & On-Premise capable Kubernetes (*kops*)
- Jenkins Pipelines for CI/CD, Self-Service
- Multi-Availability Zone, cost-effective Load Balancing
- Modern Monitoring Stack:
  - Prometheus, Grafana, CloudWatch, ElasticSearch
- Worldwide Highly Available Content Delivery Network
  - S3, CloudFront, Lambda@Edge for static-content delivery

aws partner network
Standard
Consulting Partner
Solution Provider

HashiCorp Terraform

ANSIBLE

kubernetes

Munich ReThink GmbH

ventx
we make IT

Infrastructure as Code

→ Only rely on temporary AWS credentials *(Security)*

→ Code can be used for new environments easily *(Portability)*

→ Continuous Deployment with checks and tests *(Deliverability)*

Bitbucket Webhook →  API Gateway →  Lambda →  S3 →  CodeBuild →  HashiCorp Terraform

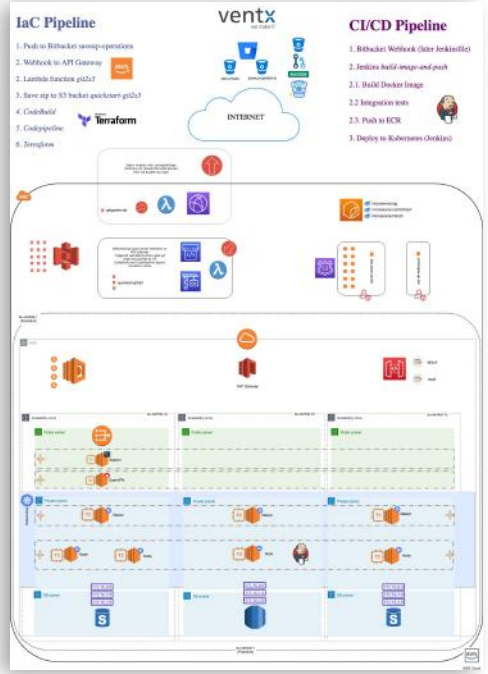
1. Bitbucket git commit Webhook → API Gateway
2. Lambda function: clone git, zip, upload → S3 bucket
3. CodeBuild/CodePipeline deploys Terraform stack & modules
    a. *Continuous Infrastructure* checks for new / updated resources
    b. Informs staff for errors *(chatOps)*
    c. Keeps infrastructure always in its desired state *(declarative)*

Munich ReThink GmbH

ventx we make IT

Why Infrastructure as Code ?

- **Declarative,** contemporary approach instead of (imperative) manual processes
  - Define desired state of the environment
  - Manual tedious processes are ineffective and error-prone
- **Single Source of Truth** in git repository
  - Represents the desired state of infrastructure
  - Shows relations, dependencies and components of resources
- **Less error-prone**
  - More automation through scriptable, consistent infrastructure
  - Infrastructure code is version-controlled and peer reviewed (PRs)
  - Faster rollback and better traceability
- **Cost-efficient**
  - Engineers spend less time with manual processes, more time for high-value tasks
  - Automatically scale in / out infrastructure as demand grows / shrinks
- **Faster execution**
  - Define infrastructure and spin up environments spanning hundreds, or even thousands of resources
  - Use multiple regions for High Availability easily



Munich ReThink GmbH

ventx
we make IT

Kubernetes *(K8s)* & Deployments

- Fully containerized, highly available microservice deployments
- Helm charts for all product microservices and internal services
- Templatable charts for Multi-Tenant deployments
- Cost-effective HA Load-Balancing with nginx-ingress
- Automatic SSL/TLS Rollout with cert-manager
- OpenVPN with Multi-Factor-Authentication *(MFA)* for secure k8s API access
- Single Point of Trust via IAM for kubectl / k8s-API access

kubernetes

Why  kubernetes ?

- De-facto Industry standard for container orchestration *(developed by Google, now CNCF)*
- Portable cloud platform *(no vendor lock-in)*
- Increased hardware utilization with fully-containerized environment
- Scalable from hundreds to thousands and even millions of containers
- High portability of software which can run on any Kubernetes cluster
- Fully modular architecture so you can customize your platform as needed
- Easy rollout & rollback
- Health-Checks *(readiness & liveness)*
- Auto-Healing and Auto-Scaling
- Load-Balancing with advanced traffic routing
- Designed for Continuous Deployment

Why CI/CD ?

- **Continuous Integration**, developers use a shared repository for their work and each commit gets verified by integration tests
  - No more “integration hell”
  - Less errors and faster problem detection
  - Higher developer efficiency
  - Deliver software more rapidly
- **Continuous Delivery**, extends the CI process for faster and more regular deployments
  - Release in small, incremental steps
  - Higher agility in development and release processes
  - Deploy to other environments (staging / production) with deployment pipelines
  - Release more often
  - Less pressure for small changes, iterate faster
  - Fast feedback for developers

Participants

- Boris Girsch
- Martin Wawrzynek
- Marcel Oed
- Hans-Jörg Wieland

Munich ReThink GmbH

ventx
we make IT