Modernizing Zurich Insurance with ROSA: Accelerating Innovation and Agility in the Cloud

15.01.2024

Simon Galbierz, Timo Bernard, Philipp Hoegner, Kevin Nash

Introduction

ZURICH

Philipp Hoegner

Zurich - Head of Cloud Center of Excellence Germany

Simon Galbierz

Zurich - Cloud Engineer

Timo Bernard

Zurich - Cloud Engineer

Kevin Nash

AWS - Senior Solutions Architect

Agenda

ZURICH®

1. About Zurich Insurance Germany and our Container First strategy with Red Hat Openshift Service on AWS (ROSA)
2. Modernization and innovation with containers on AWS
3. Zurich Insurance Germany ROSA setup
4. GitOps for declarative cluster and application configuration
5. Secure and compliant application operation with Istio Service Mesh and 4-eye principle

About Zurich Insurance Germany

ZURICH

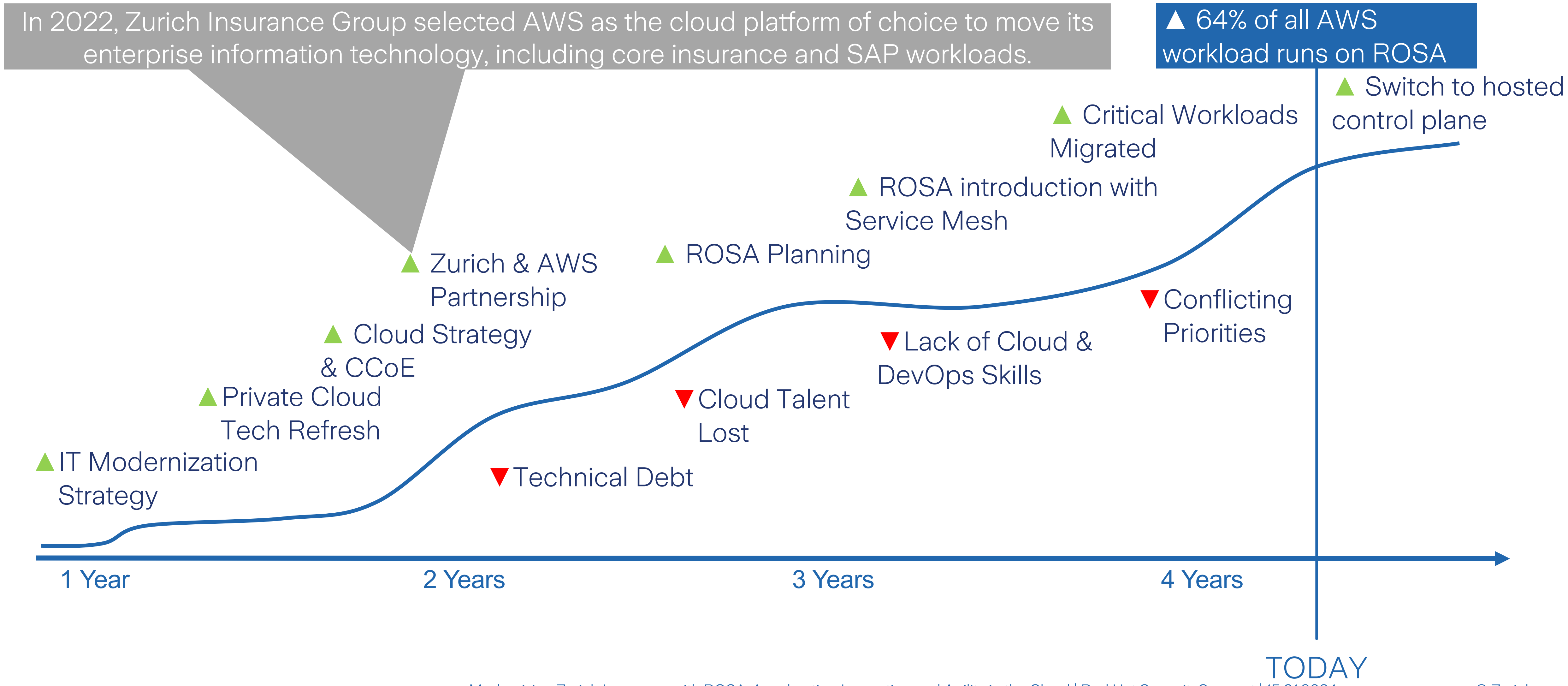
One global Zurich Insurance Group with long-standing and strong presence in Germany with over 6 million customers

- Around 4,500 employees at two main sites (Cologne and Frankfurt am Main)
- Fiscal Year 2023 insurance premiums of EUR 6bn; investments of EUR 49bn
- 3 Brands: ZURICH | Real Garant Versicherung AG, Ein Unternehmen der ZURICH Gruppe | DA direkt
- 34/40 DAX companies are insured with Zurich
- Top 10 insurance company in Germany; Top 2 provider of unit-linked Life insurance in Germany
- Top positions in employer rankings in Germany with outstanding employee satisfaction

Cologne
Frankfurt
ZURICH
Capital
FOCUS TOP NATIONALER ARBEITGEBER 2024
ZURICH
TOP ARBEITGEBER 2024 FOCUS DIVERSITY

Our Cloud and Container Journey

ZURICH



In 2022, Zurich Insurance Group selected AWS as the cloud platform of choice to move its enterprise information technology, including core insurance and SAP workloads.

▲ 64% of all AWS workload runs on ROSA

▲ IT Modernization Strategy

▲ Private Cloud Tech Refresh

▲ Cloud Strategy & CCoE

▲ Zurich & AWS Partnership

▼ Technical Debt

▲ ROSA Planning

▼ Cloud Talent Lost

▲ ROSA introduction with Service Mesh

▼ Lack of Cloud & DevOps Skills

▲ Critical Workloads Migrated

▼ Conflicting Priorities

▲ Switch to hosted control plane

Our mission: Secure Value Creation and Innovation

ZURICH

Financial services need to balance efficiency, innovation and compliance

Cloud First is Key

- Not solely for infrastructure modernization, but as a catalyst for our transformation
- For increased value creation and innovation

Adoption of Containers with ROSA

- Enable cloud native architecture through simplified application modernization
- Increase security posture
- Ensure usage of vendor agnostic technology to adhere to regulatory requirements (e.g. DORA)
- Decrease management overhead and improve reusability while accelerating cloud adoption

Modernization & Innovation

CIOs say that 80% of developers' time is spent on the operations and maintenance of applications and only 20% of the time is actually spent on innovation

Source: Deloitte 2019

Why AWS?

AWS Availability Zone Design

34 AWS Regions
108 AWS Availability Zones

TCO Price Reductions

134 Price Reductions
Since 2006

Breadth & Depth

200+ fully featured services support any cloud workload

Hardware Innovation

Recognized as the Most Innovative Cloud Provider; Exclusive Purpose-Built Hardware

AWS Inferentia and AWS Trainium

AI chips built for price performance of model training & serving

AWS Graviton

Powerful & efficient custom built Arm-based processors

AWS Nitro System

Reduces costs of virtualization & maximizes server resources delivered to your EC2 instances

Amazon CloudFront

600+ Points of Presence, 100+ cities, 50+ countries

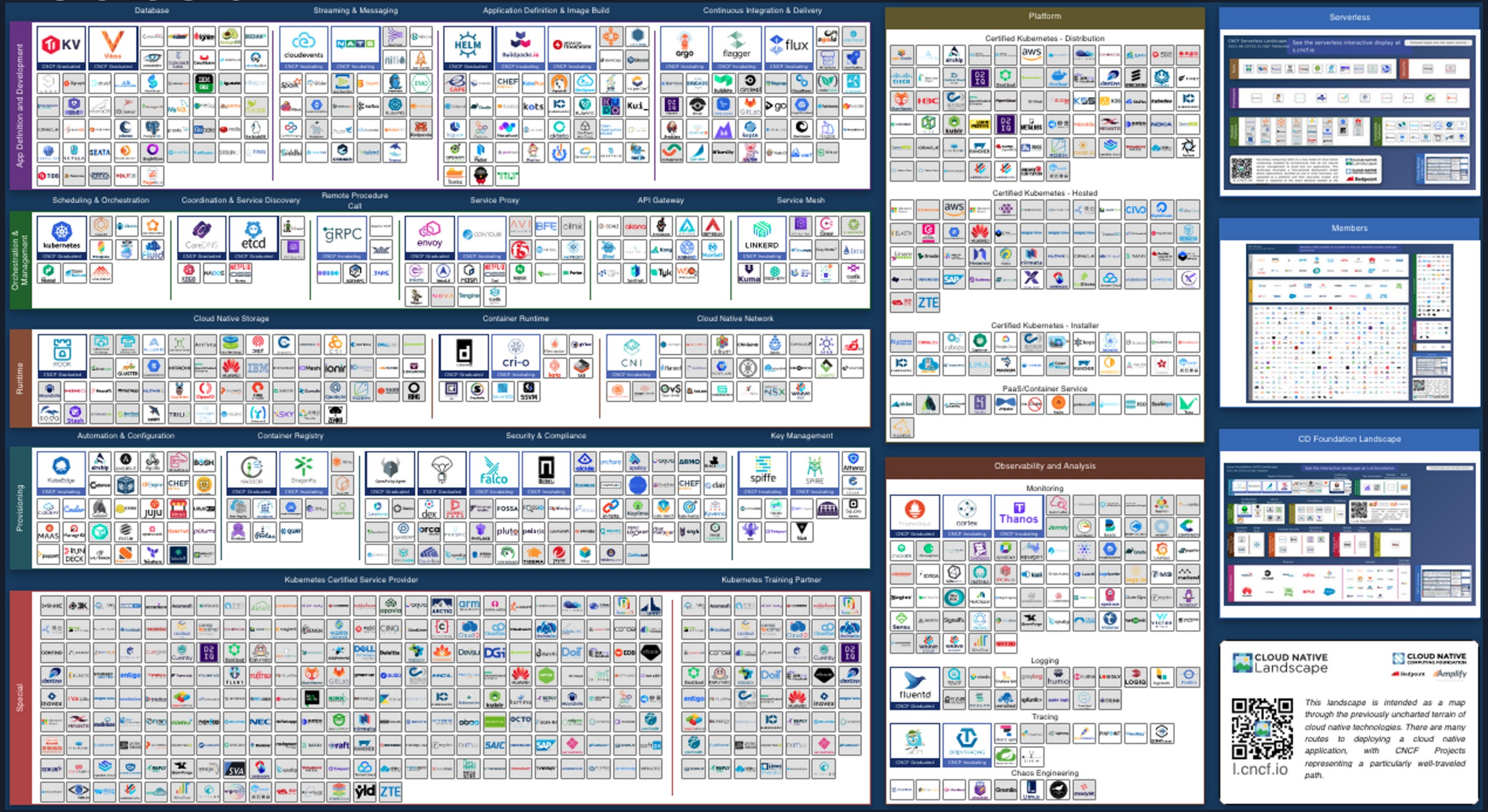
Resilient Infrastructure

Gartner Magic Quadrant Worldwide Leader for Cloud Infrastructure as a Service

Sustainable Operations

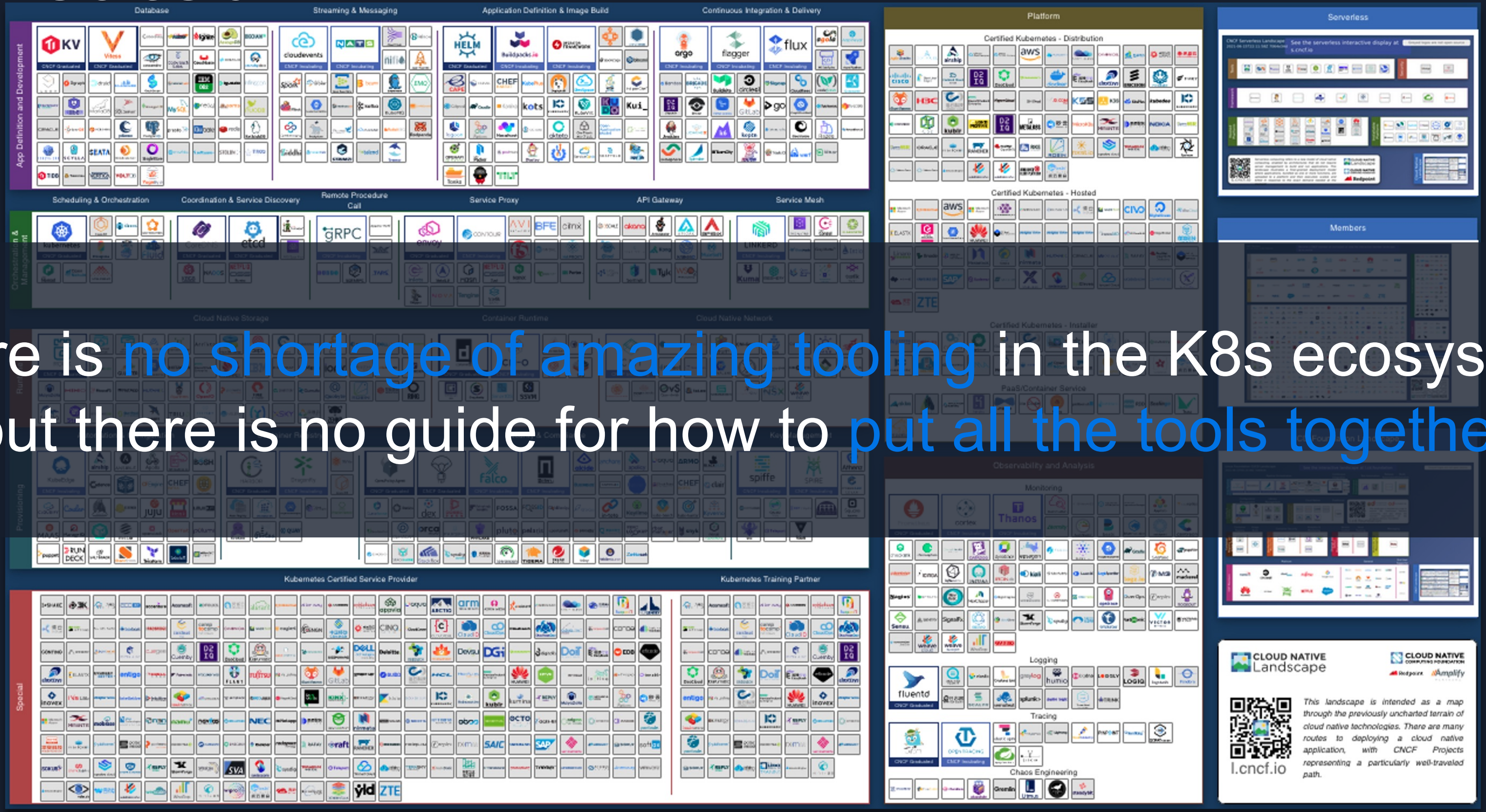
Match 100% of the electricity we consume with renewable energy

The containers landscape is vast & complicated


Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
App Definition and Development
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
Cloud Native Storage
Container Runtime
Cloud Native Network
Runtime
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Provisioning
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Members
CD Foundation Landscape
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
l.cncf.io

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

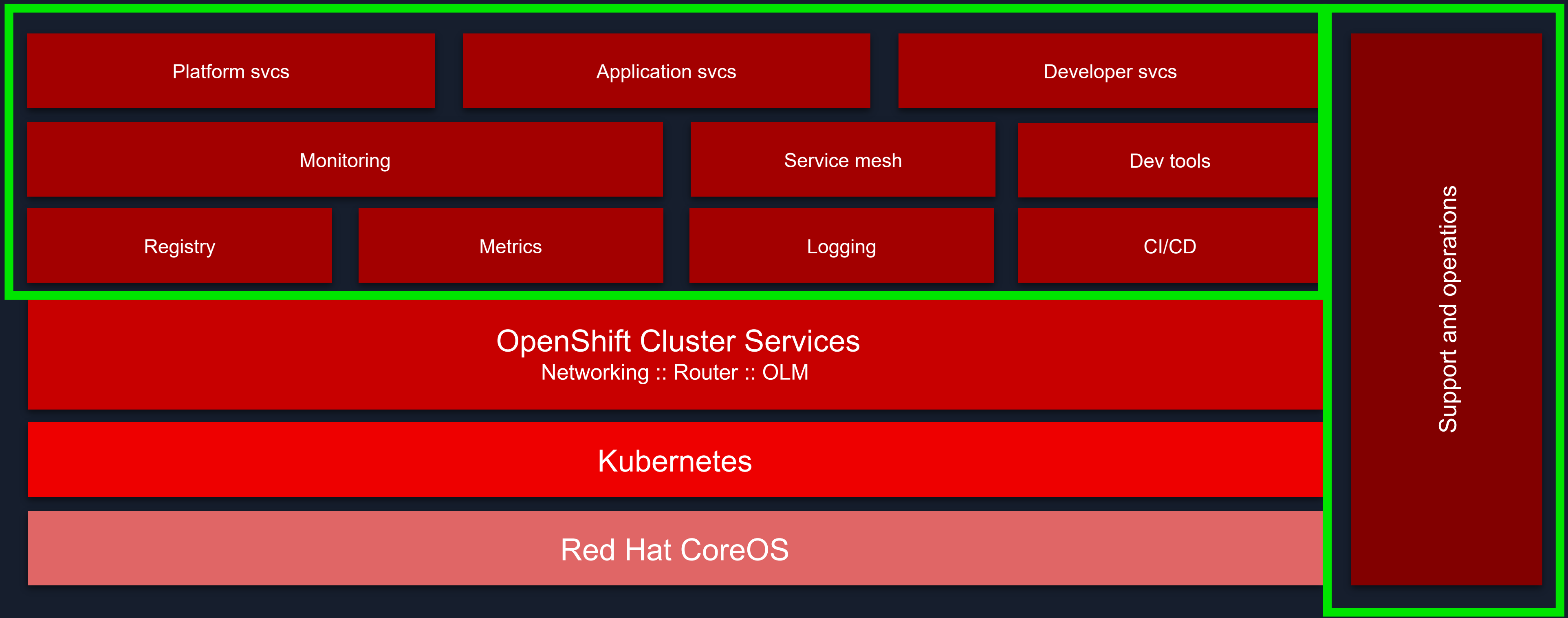
The containers landscape is vast & complicated


Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
App Definition and Development
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Serverless
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
Members
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io

There is no shortage of amazing tooling in the K8s ecosystem, but there is no guide for how to put all the tools together

ROSA : The turnkey Kubernetes platform

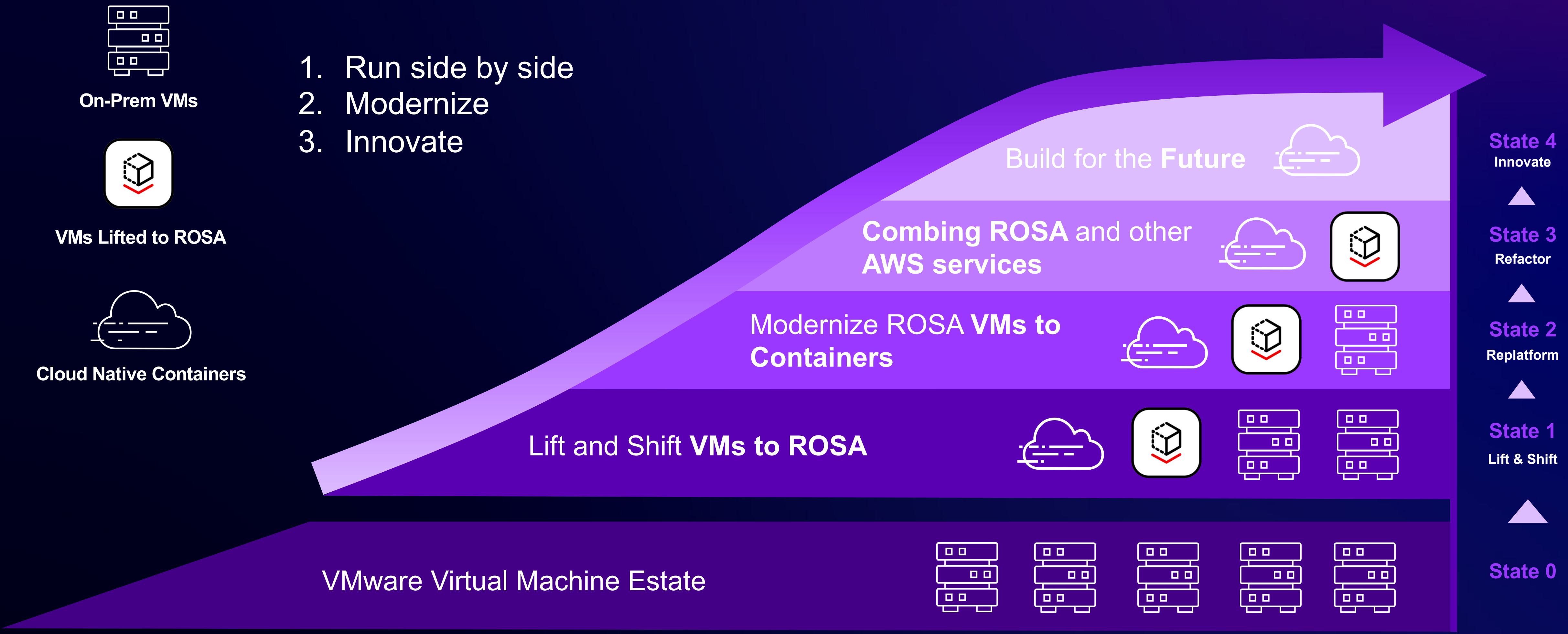
- No assembly required
- Managed platform stack
- Opinionated defaults
- Supported set of integrations



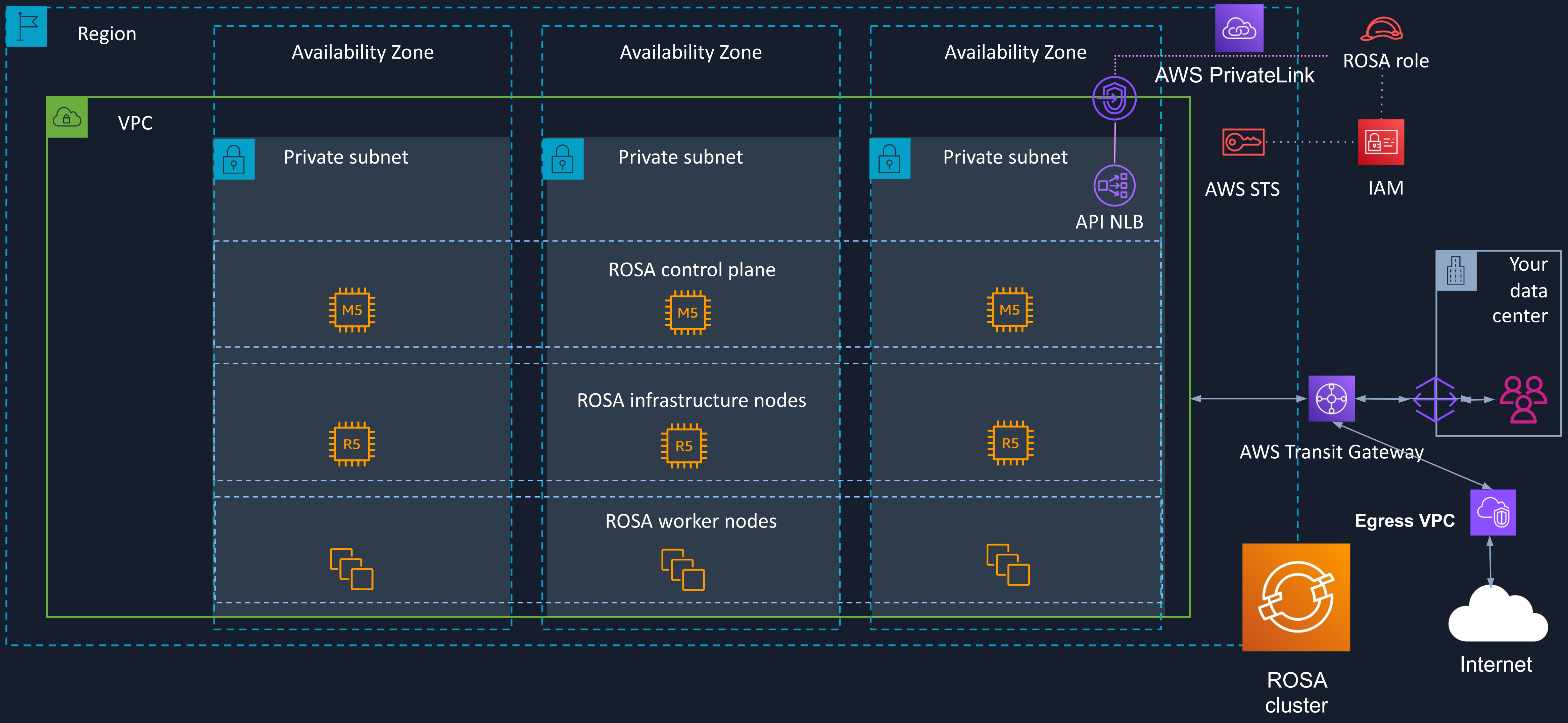
Virtual Machine Modernization Journey

WITH OPENSHIFT VIRTUALIZATION, VMS ARE JUST ANOTHER WORKLOAD MANAGED BY ROSA

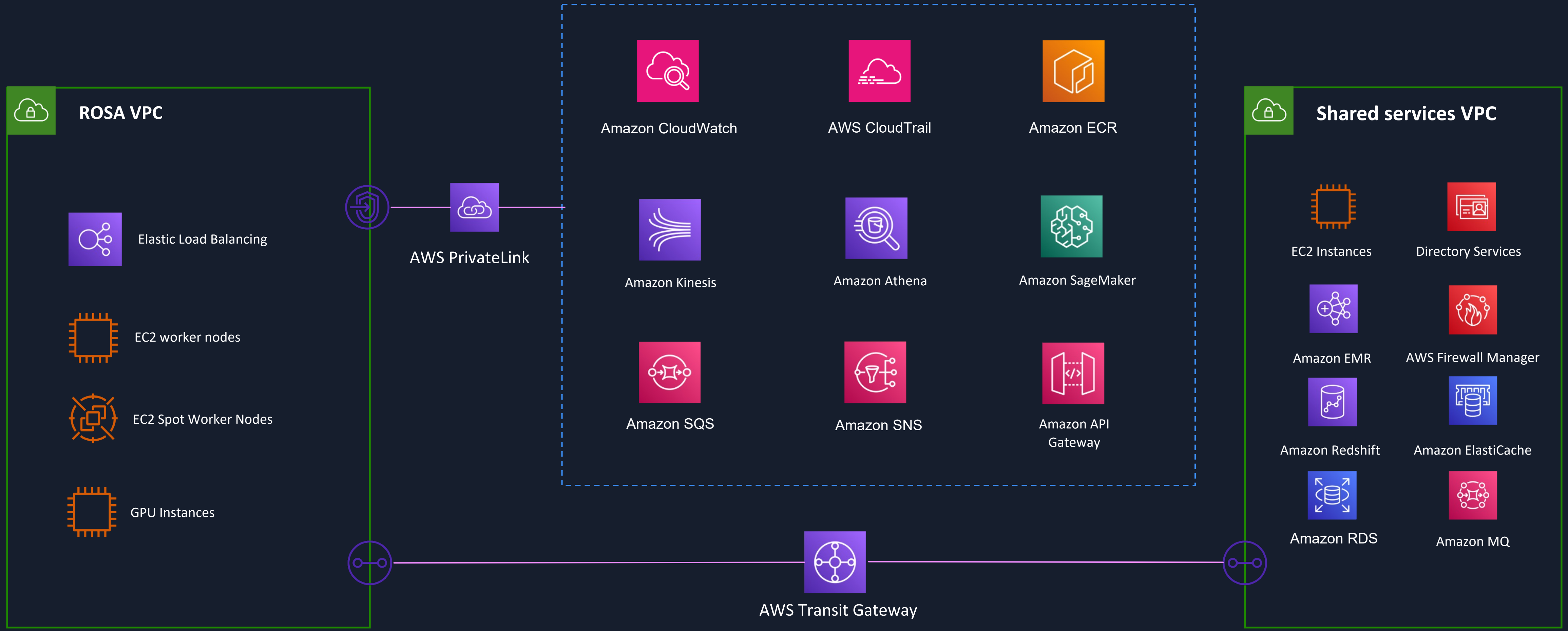
1. Run side by side
2. Modernize
3. Innovate



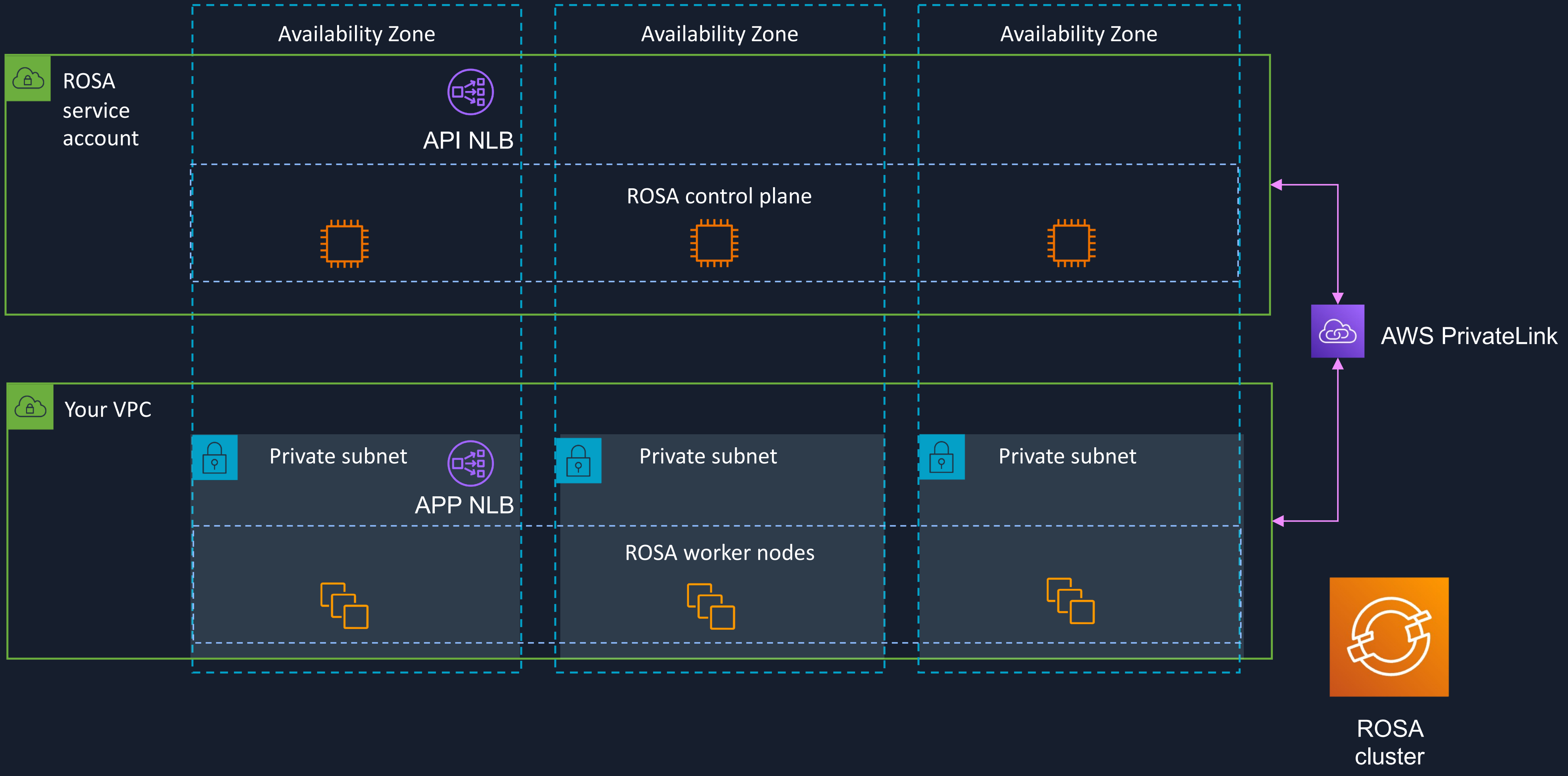
ROSA classic multi-AZ AWS PrivateLink cluster



Secure access to AWS services

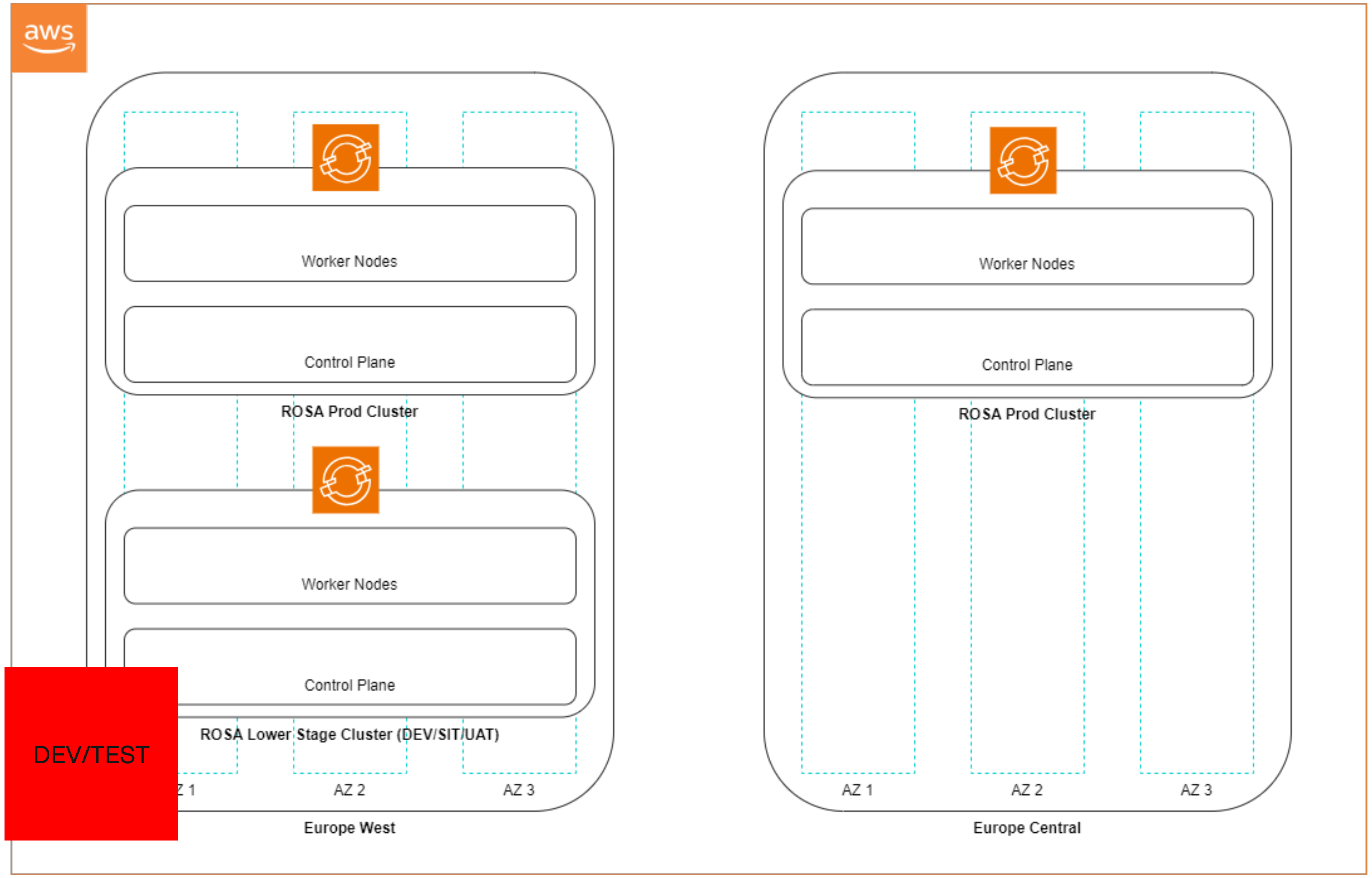


ROSA with hosted control planes (HCP)



Zurich Insurance Germany ROSA Setup

Staging Concept with identical configuration



Stage Separation

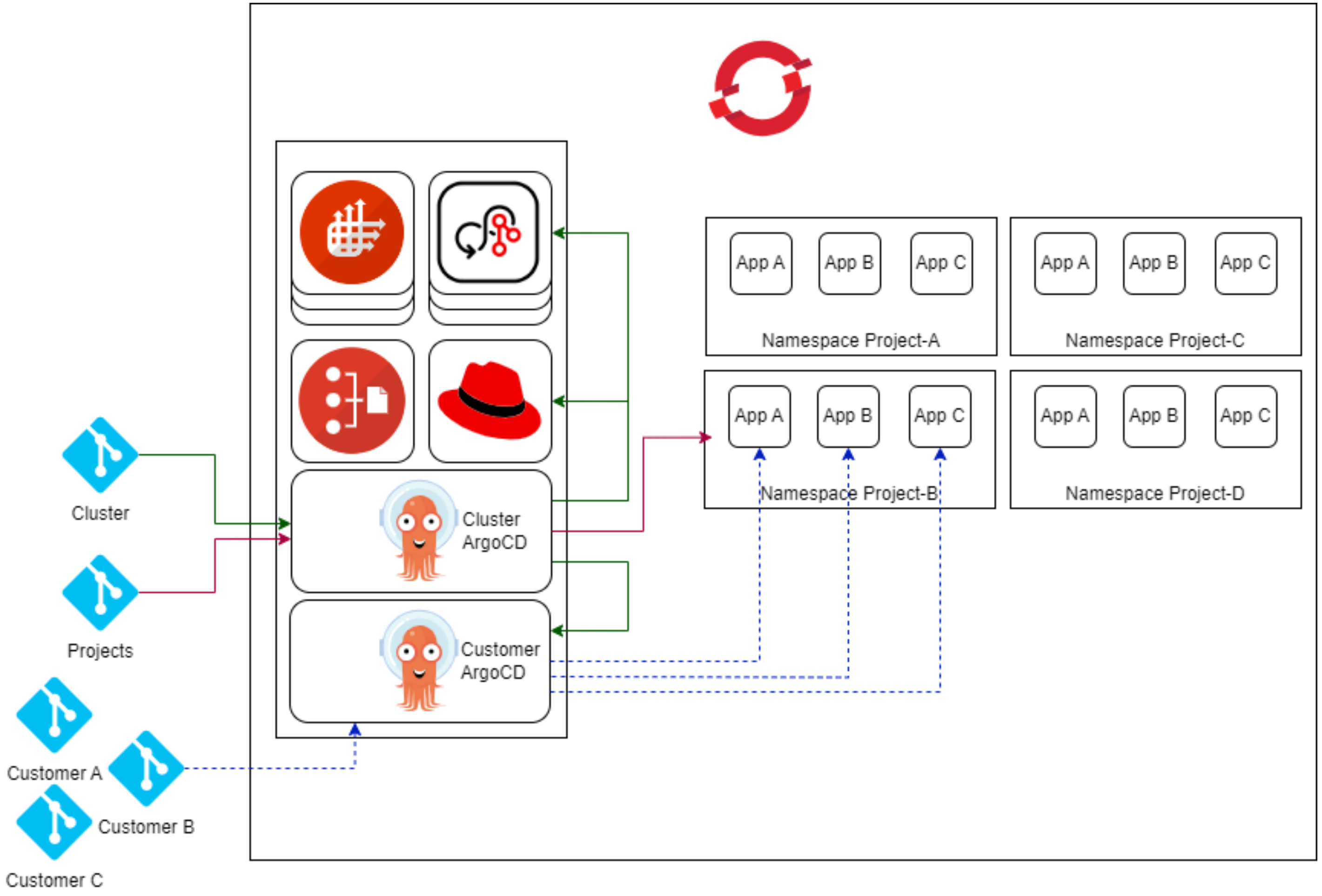
- Seperation of lower and upper stages through dedicated clusters
  - increased regulatory compliance
  - increased cost efficiency
  - trust in application stability, through thorough testing in lower stages
- Identical configuration across stages through IaC for reproducability and consistency

Redundant Production Clusters

- Nearly equal distribution of workload across regions to increase resilience in case of disasters
- All clusters are setup with dynamic autoscaling to improve efficiency

GitOps for Declarative Cluster and Application Configuration

Separation of administration and project related activities



Cluster ArgoCD

- Oversees and manages all cluster-related configurations
- Administers customer ArgoCD instances and their namespaces
- Serves as a centralized control center to streamline maintenance and updates with minimal effort

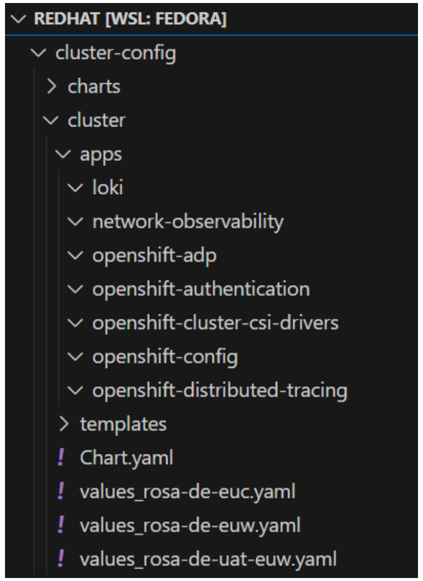
Customer ArgoCD

- Enables our customers to create their own applications within defined boundaries
- Implements separate RBAC configurations per namespace, ensuring isolation while using a shared ArgoCD instance

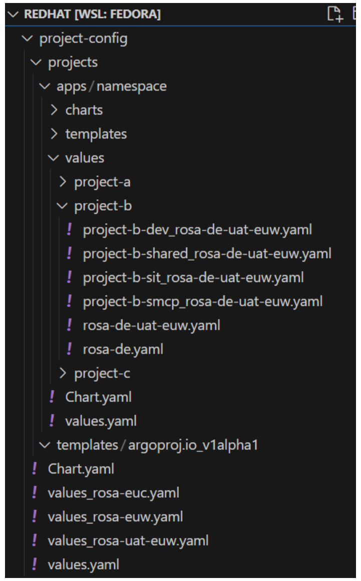
GitOps for Declarative Cluster and Application Configuration

Distinct Code Repositories

- Two individual repositories
  - Cluster Definition
  - Project Definition
- Branching for lower stages and production
  - Merge only on release for production



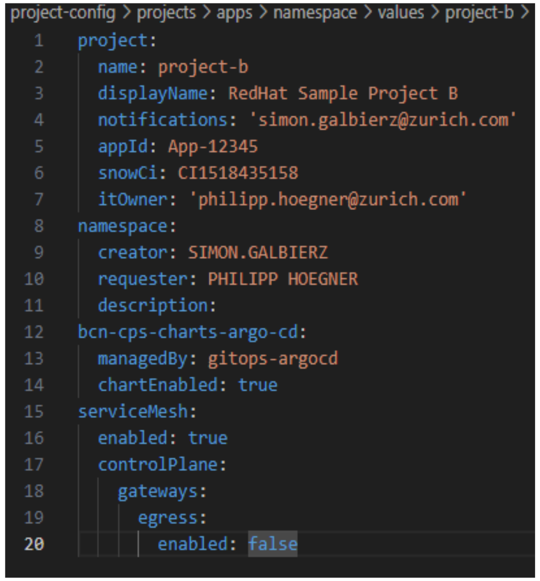
```
REDHAT [WSL: FEDORA]
  cluster-config
    charts
    cluster
      apps
        loki
        network-observability
        openshift-adp
        openshift-authentication
        openshift-cluster-csi-drivers
        openshift-config
        openshift-distributed-tracing
      templates
      Chart.yaml
      values_rosa-de-euc.yaml
      values_rosa-de-euw.yaml
      values_rosa-de-uat-euw.yaml
```



```
REDHAT [WSL: FEDORA]
  project-config
    projects
      apps / namespace
        charts
        templates
        values
          project-a
          project-b
            project-b-dev_rosa-de-uat-euw.yaml
            project-b-shared_rosa-de-uat-euw.yaml
            project-b-sit_rosa-de-uat-euw.yaml
            project-b-smcp_rosa-de-uat-euw.yaml
            rosa-de-uat-euw.yaml
            rosa-de.yaml
          project-c
        Chart.yaml
        values.yaml
      templates / argoproj.io_v1alpha1
    Chart.yaml
    values_rosa-euc.yaml
    values_rosa-euw.yaml
    values_rosa-uat-euw.yaml
    values.yaml
```

General Project Definition

- General project overview
- Provides change approver information for the project and relevant stakeholders in case of any problems, i.e.
  - Platform incidents
  - Application related incidents
- Internal references to the central CMDB



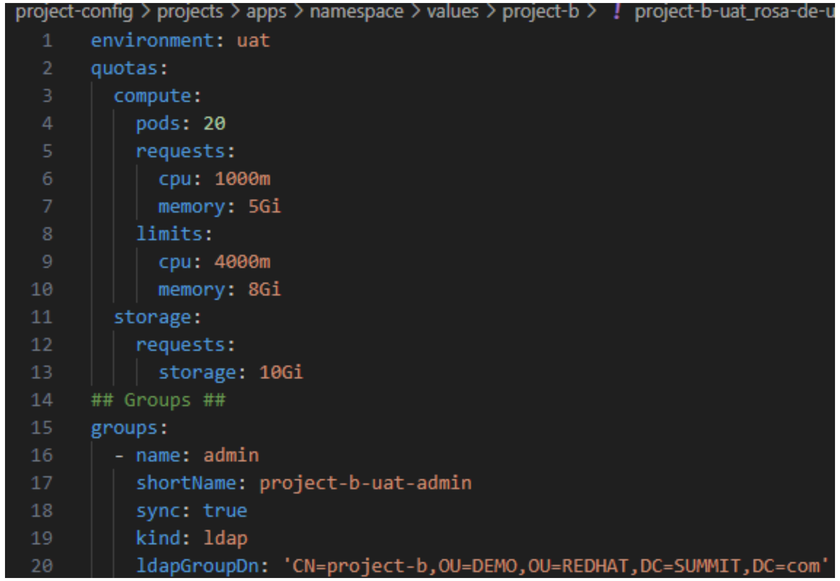
project-config > projects > apps > namespace > values > project-b >

```yaml
project:
  name: project-b
  displayName: RedHat Sample Project B
  notifications: 'simon.galbierz@zurich.com'
  appId: App-12345
  snowCi: CI1518435158
  itOwner: 'philipp.hoegner@zurich.com'
namespace:
  creator: SIMON.GALBIERZ
  requester: PHILIPP HOEGNER
  description:
bcn-cps-charts-argo-cd:
  managedBy: gitops-argocd
  chartEnabled: true
serviceMesh:
  enabled: true
  controlPlane:
    gateways:
      egress:
        enabled: false
```

GitOps for Declarative Cluster and Application Configuration

Namespace Definition

- Defines the inner boundaries of the namespace
- Customer must supply us with information of their expected workload
- Defines the Active Directory groups that will be linked to the RBAC’s

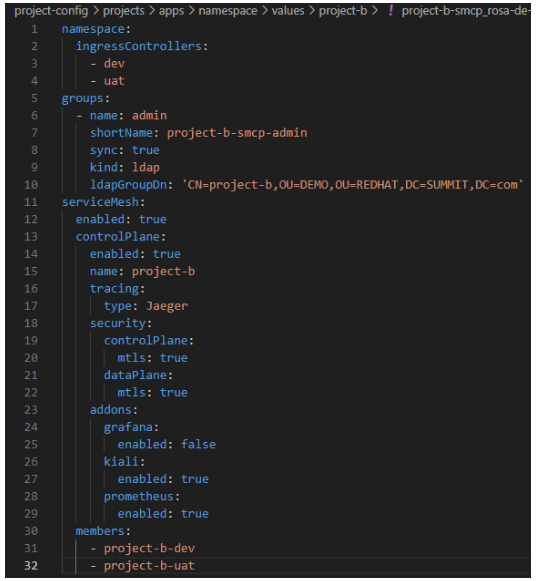


project-config > projects > apps > namespace > values > project-b > ! project-b-uat_rosa-de-u

```yaml
environment: uat
quotas:
  compute:
    pods: 20
    requests:
      cpu: 1000m
      memory: 5Gi
    limits:
      cpu: 4000m
      memory: 8Gi
  storage:
    requests:
      storage: 10Gi
## Groups ##
groups:
  - name: admin
    shortName: project-b-uat-admin
    sync: true
    kind: ldap
    ldapGroupDn: 'CN=project-b,OU=DEMO,OU=REDHAT,DC=SUMMIT,DC=com'
```

SMCP Definition

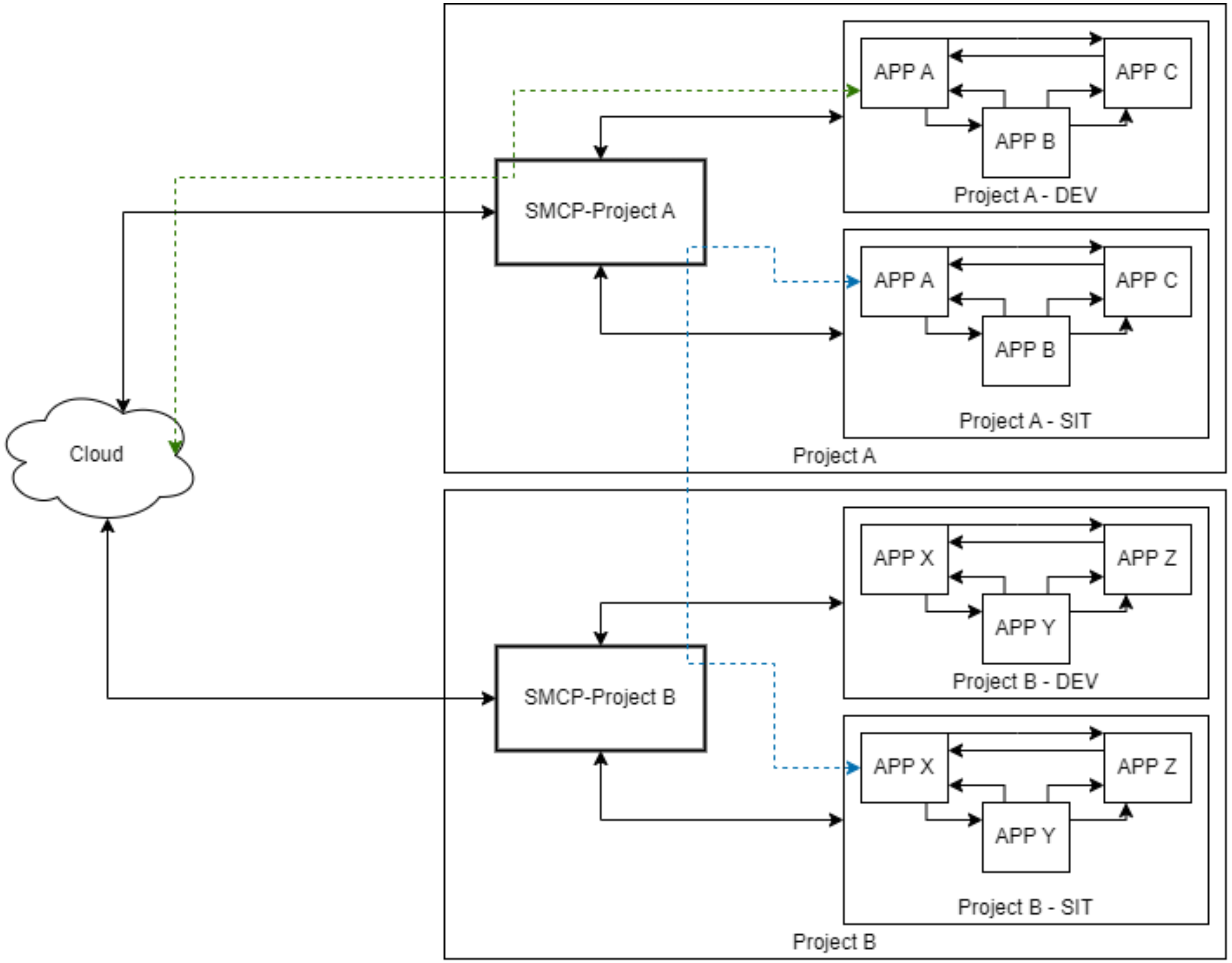
- Configures the Service Mesh Control Plane (SMCP) namespace for this project
- Controls the ingress and if it can be accessed from the internet
- Deploys the tool stack with Kiali/Jäger/Prometheus for easy monitoring and debugging
- Enforces mTLS



project-config > projects > apps > namespace > values > project-b > ! project-b-smcp_rosa-de-

```yaml
namespace:
  ingressControllers:
    - dev
    - uat
groups:
  - name: admin
    shortName: project-b-smcp-admin
    sync: true
    kind: ldap
    ldapGroupDn: 'CN=project-b,OU=DEMO,OU=REDHAT,DC=SUMMIT,DC=com'
serviceMesh:
  enabled: true
  controlPlane:
    enabled: true
    name: project-b
    tracing:
      type: Jaeger
    security:
      controlPlane:
        mtls: true
      dataPlane:
        mtls: true
    addons:
      grafana:
        enabled: false
      kiali:
        enabled: true
      prometheus:
        enabled: true
  members:
    - project-b-dev
    - project-b-uat
```

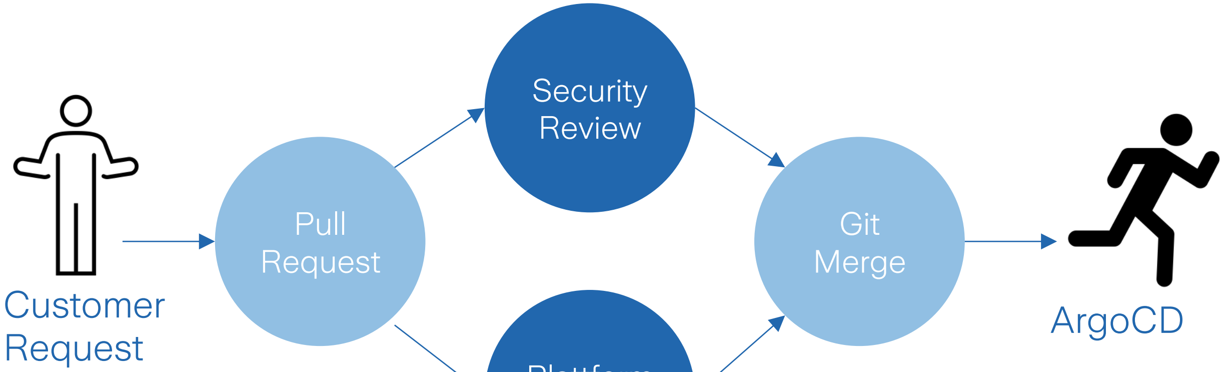
Secure and Compliant Application Operation with Istio Service Mesh and 4-eye Principle



Namespace Isolation

- Allow communication inside Namespace
- Enforcement of internal communication via SideCar w/ mTLS
- Blackhole by default for outgoing communication
  - Whitelisted external communication through the SMCP only
- Incoming communication via route or joined SMCP‘s

Secure and Compliant Application Operation with Istio Service Mesh and 4-eye Principle



Revision Process for SMCP Configuration

- Team creates pull request with SMCP configuration
- Plattform team reviews request and checks syntax
- Security team reviews pull request against defined security standards
- Everything-as-code (Git)
  - Defined approval flow for merges into central repository
  - Change record of requests and approvals

Summary & Outlook

1. ROSA enables us to **balance efficiency, innovation and compliance**
2. We are convinced **Container and ROSA heavily improve our security & compliance** posture resulting from standardization and automation
3. The **operational workload is heavily reduced** through abstraction and everything-as-code
4. We experienced **cost reduction through containerization and the usage of ROSA** and its opportunity to scale according to our business needs

Join us for a ROSA Workshop

A hands-on experience

Zurich 04.02.25