Konecta ERP Cloud – Architecture, Changes, and Readiness Report

Scope

This document summarizes all Terraform edits applied in this session, describes the current single-region architecture, how components communicate, and what remains for a seamless deployment and multi-region expansion. Structure mirrors the repo hierarchy where helpful.

---

1) Changes by Directory/Module

terraform/

- `main.tf`
  - Wires modules (`vpc`, `rds`, `eks`, `cloudwatch`, `ecr`, `cloudfront`).
  - Passes `ssl_certificate_arn` to `eks` for HTTPS on ALB and HTTP→HTTPS redirect.
  - Integrates `cloudwatch` module for RDS/EKS dashboards and alarms.
  - Keeps NAT enabled and exposes variables for optional VPC peering.
- `variables.tf`
  - Added inputs for: `rds_alarm_actions`, `ssl_certificate_arn`, VPC peering (`enable_vpc_peering`, `peer_vpc_id`, `peer_cidr_block`, `peer_region`).

terraform/modules/vpc/

- `main.tf`
  - VPC with DNS support; public and private subnets across two AZs.
  - NAT Gateway for private egress (assumed on; no VPC endpoints now).
  - Subnet tags for EKS/ELB controller:
    - Public: `kubernetes.io/role/elb = 1`
    - Private: `kubernetes.io/role/internal-elb = 1`
    - All: `kubernetes.io/cluster/konecta-erp-${environment} = shared`
  - Optional VPC Peering (requester side) with routes to peer CIDR.
- `variables.tf`
  - `enable_nat_gateway` (default true).
  - Optional peering variables: `enable_vpc_peering`, `peer_vpc_id`, `peer_cidr_block`, `peer_region`.
- `outputs.tf`
  - Standard VPC outputs and optional `vpc_peering_connection_id`.

terraform/modules/eks/

- `main.tf`

    - EKS cluster (Fargate-only) with control plane logging enabled: `api`, `audit`, `authenticator`, `controllerManager`, `scheduler`.
    - Fargate pod execution role with `AmazonEKSFargatePodExecutionRolePolicy` and `CloudWatchAgentServerPolicy`.
    - OIDC/IRSA provider created (tls thumbprint) for service accounts.
    - IRSA role for CloudWatch agent (`aws-observability/cloudwatch-agent`).
    - EKS Add-on: `amazon-cloudwatch-observability` installed via Terraform.
    - ALB resources: SG open on 80/443, ALB+TG with health checks, listener 80 redirect → 443 when `ssl_certificate_arn` set, listener 443 terminates TLS.

- `variables.tf`

    - Added `ssl_certificate_arn` for HTTPS listener.

- `outputs.tf`

    - Cluster name, endpoint, ALB DNS/TG, Fargate outputs.

terraform/modules/cloudwatch/

- `main.tf`

    - RDS dashboard and alarms (CPU, FreeStorage, Connections).
    - EKS (Fargate-only) Container Insights dashboards and alarms using pod-level metrics (no NodeName dependency).
    - Control plane log group with retention.

- `variables.tf`

    - Inputs for environment, region, RDS identifier, EKS cluster name, `rds_alarm_actions`, `eks_alarm_actions`.

terraform/modules/rds/

- `main.tf`

    - PostgreSQL RDS with private subnets, SG allowing VPC CIDR (extensible via `additional_allowed_cidrs`).
    - Backups: `backup_retention_period = 7`, `copy_tags_to_snapshot = true`.
    - Protection/observability: `deletion_protection`, `enabled_cloudwatch_logs_exports`, Enhanced Monitoring role (interval 60s), Performance Insights (7 days).

- `variables.tf`

    - `additional_allowed_cidrs` (to allow access from peered secondary VPC).
    - Switches for Enhanced Monitoring/PI.

- `outputs.tf`

    - Added `db_arn`, `db_identifier` for cross-region replica creation.

terraform/modules/ecr/

- ECR repositories for services and lifecycle policies (keep last 10 images).
  - IAM policy attached to node role in earlier iteration was replaced with a self-contained ECR module that outputs repo URLs (node/fargate pulls use default ECR permissions plus task role configuration).

terraform/modules/cloudfront/

- CloudFront distribution fronting ALB (HTTPS), ready for WAF/logging; S3 origin removed (frontend will be served from EKS).

---

2) Current Architecture (Single Region)

- Networking
  - One VPC with two public and two private subnets across AZs.
  - Public subnets: ALB. Private subnets: EKS Fargate pods and RDS.
  - NAT Gateway provides egress for private subnets.
- EKS (Fargate-only)
  - Control plane with public/private access (restricted later as needed).
  - Fargate profile in private subnets for workloads.
  - IRSA enabled for add-ons and future service accounts.
  - CloudWatch Observability add-on enabled (Container Insights for Fargate).
- Ingress
  - Application Load Balancer in public subnets.
  - HTTP listener redirects to HTTPS when `ssl_certificate_arn` is set; HTTPS listener forwards to target group.
  - Kubernetes ALB integration will be managed by DevOps via aws-load-balancer-controller Helm chart (IRSA prerequisites satisfied, subnets tagged).
- Data
  - RDS PostgreSQL in private subnets; encrypted; 7-day automated backups; PI + Enhanced Monitoring.
  - SG allows VPC CIDR; extendable to peer VPC CIDR for multi-region.
- Observability
  - CloudWatch dashboards/alarms for RDS and EKS (Fargate pod metrics).
  - EKS control-plane logs to CloudWatch Logs.
- Edge
  - CloudFront in front of ALB for global HTTPS, security, and future multi-origin/failover.

Communication flows

- User → CloudFront (HTTPS) → ALB (HTTPS) → EKS Service/Ingress → Fargate Pods.
- Fargate Pods → RDS (5432) inside VPC private subnets.
- Fargate Pods → Internet egress via NAT for ECR pulls/updates.

---

3) Readiness Report (Single Region)

Ready

- VPC/subnets/NAT and subnet tags for ELB integration.
- EKS Fargate cluster with IRSA and CloudWatch Observability add-on.
- ALB with optional HTTPS and HTTP→HTTPS redirect.
- RDS private + backups + monitoring + PI + enhanced monitoring.
- CloudWatch dashboards/alarms for RDS and EKS.
- CloudFront in front of ALB for HTTPS/global edge.

Pending for a seamless deployment (expected from DevOps)

- Install aws-load-balancer-controller via Helm/manifests, using IRSA (role can be added here if desired) and proper annotations on Services/Ingress.
- Deploy application charts/manifests, configure Services/Ingress, ensure targetType=ip for Fargate.
- Provide `ssl_certificate_arn` (ACM) and set Route53 records (if using custom domains).
- Confirm app retrieves DB credentials (Secrets Manager or env vars) and can connect to RDS.

Optional hardening/tuning

- Restrict RDS SG to an SG used by Fargate ENIs instead of VPC CIDR.
- Prefer private EKS endpoint + controlled public CIDR list.
- Tune CloudWatch thresholds/retention.

---

4) Multi-Region Readiness (What's in place vs what remains)

In place

- Optional VPC peering (requester side) and routes from primary to peer CIDR.
- RDS primary configured as a valid source for cross-region read replica (backups on, encryption, snapshot tag copy, observability).
- CloudFront in front of ALB enabling future multi-origin failover strategy.

To add in secondary region

- Mirror: VPC/subnets/NAT, EKS (Fargate), OIDC/IRSA, ALB, CloudWatch Observability.
- Accept VPC peering and add reciprocal routes; if needed, Route53 Resolver rules for cross-VPC DNS.
- RDS cross-region read replica resource; add `ReplicaLag` alarms; promotion automation (Lambda/runbook) to meet RTO/RPO.
- CloudFront: add secondary ALB as a second origin with failover/routing policy; WAF optional.
- Secrets/KMS: replicate secrets and ensure KMS keys available in region.

RTO/RPO note

- Targets (RTO 4h / RPO 1h) require the replica, promotion automation, and pre-provisioned (or rapidly provisionable) secondary stack.

---

5) Action Items Summary

- DevOps
  - Install aws-load-balancer-controller and configure Ingress/Services.
  - Deploy application workloads, wire Secrets, set DNS (Route53), and ACM.
- Platform (optional follow-ups via Terraform)
  - Add IRSA role/policy for aws-load-balancer-controller (if you want IAM managed here).
  - Implement RDS replica in secondary region and peering accepter/routes.
  - Extend CloudFront to multi-origin failover.

---

6) Quick Verification Checklist

- `terraform validate` and `terraform plan` succeed.
- EKS is active; CloudWatch Observability add-on shows pod metrics in Container Insights.
- RDS reachable from a test pod (`psql`), credentials resolved from Secrets.
- ALB DNS reachable; if `ssl_certificate_arn` is set, HTTP→HTTPS redirect works.
- CloudWatch dashboards show RDS and EKS metrics; alarms are in OK state.