AKKA

TECHNICAL OVERVIEW

Bring Your Own Cloud

AKKA

Bring Your Own Cloud

Technical overview

Introduction

Akka is a platform for building and running applications designed to deliver guaranteed resilience and achieve up to 99.9999% availability. This level of availability is accomplished by running applications concurrently across multiple regions, with seamless, transparent data replication between regions to ensure consistency and fault tolerance.

Akka Bring Your Own Cloud (BYOC) regions can be deployed within major cloud providers, including AWS, GCP, and Azure. These regions operate entirely within the customer's Virtual Private Cloud (VPC/VNet), ensuring customers maintain custodial ownership of the infrastructure and the environment where the Akka applications are executed. Akka BYOC regions are installed, monitored, and updated by the Akka team, enabling organizations to benefit from Akka's operational expertise while maintaining full control over their cloud environments.

This approach provides greater flexibility, compliance alignment, and cost optimization while maintaining the core benefits of Akka.

1. **Cloud-agnostic deployment**
    a. Supports major cloud providers: AWS, Azure, Google Cloud Platform (GCP)
    b. Akka services can be deployed across multiple cloud providers
2. **Enhanced security and compliance**
    a. Data remains within the client's chosen environment, aiding compliance with regional regulations (e.g., GDPR, HIPAA).
    b. Clients maintain control over their security configurations while benefiting from our platform's built-in security features.
3. **Cost management**
    a. Allows clients to utilize existing cloud contracts and pricing models.
    b. Reduces the need for data transfer and egress fees often associated with external cloud-to-cloud interactions.

AKKA

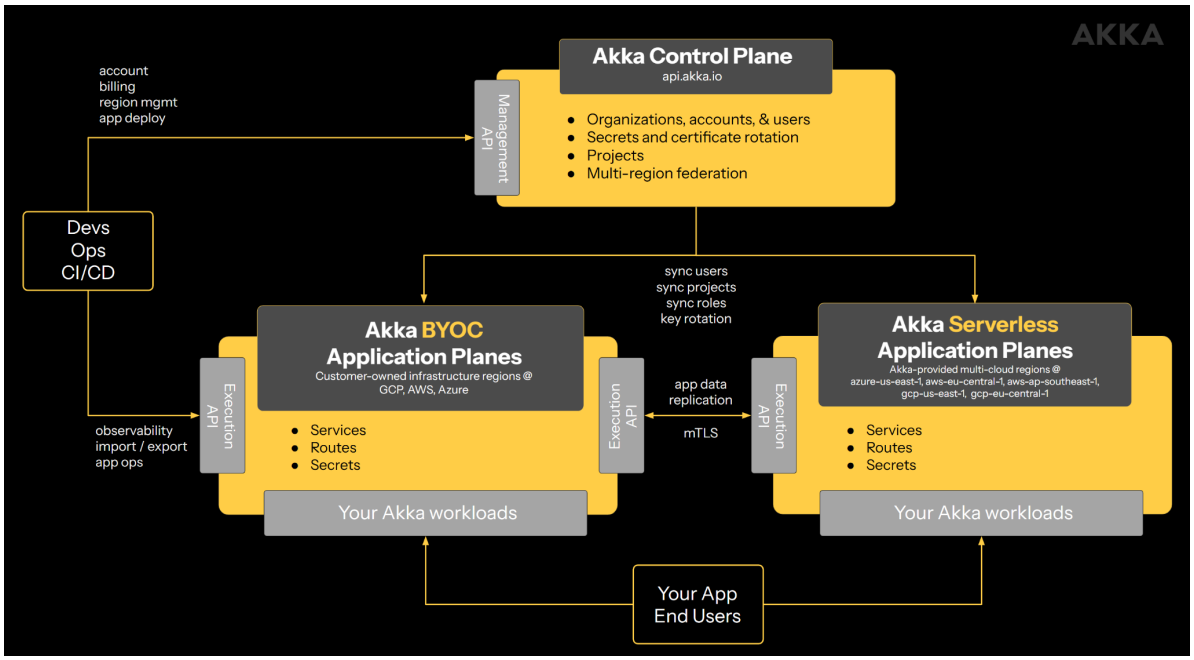
Table of Contents

Terminology Used Throughout This Document

Term	Description
Akka Application	An application that is built using the Akka SDK. Applications contain APIs, workflows, streaming consumers, timers, and views for querying data. Applications are packed into Docker images and deployed as microservice instances within an Akka operating environment. Akka applications act as their own in-memory, durable database. They take responsibility for persisting their own state. Akka apps also "cluster from within", creating a runtime cluster with other instances that handle balancing traffic, sharding data, and replicating their data to instances running within another region. Applications can be replicated between regions in different application planes if needed.
Akka SDK	Software Development Kit with support for programming components, a local debugging console, and a testkit for building, testing, and packing Akka applications.
Akka Application Plane	The runtime environment for hosting Akka applications within one or more regions. The Akka application plane provides compute, storage and I/O to execute Akka apps. It also provides automation to increase or decrease application instance capacity, observability for monitoring and debugging application behavior, and infrastructure management. The application plane is responsible for ensuring an Akka application meets its SLA by managing the Akka application and the underlying infrastructure. An Akka application plane embeds and manages: 1. A Kubernetes cluster. 2. An encrypted application data store. 3. A set of Akka operators for elasticity, observability, routing, and service management. 4. Proxies for traffic steering between regions. 5. Physical compute and storage for application execution. 6. An image repository that contains the packed applications available for deployment within that region.
Akka Control Plane	A global coordination point that is responsible for organizations, accounts, billing, and federating multiple Akka regions into a single substrate for which Akka applications can be deployed into. The Akka Control Plane runs at Akka.io and is managed by Akka. The Akka Control Plane is necessary for deployment of Akka applications while the application plane is responsible for your application's elasticity, availability, and resilience. An unavailable control plane will not impact running Akka applications.
Akka CLI	The Command Line Interface for developers, operators, and Infosec teams to interface with various Akka environments. The CLI provides utilities for building, testing, packing, and deploying Akka applications. It also provides utilities for observability, secrets management, service scaling, and account management.
Data persistence	Akka provides a durable event store for persisting data, with all data encrypted at rest. The data is stored within a journal, adhering to Akka's event-sourcing architecture. This format is optimized for Akka's internal processing and cannot be directly queried.

AKKA

Akka Control and Application Plane



Akka Control Plane

The Akka Control Plane serves as the central coordination system for managing organizational and operational aspects of Akka, such as account setup, role assignments, user access control, key rotation, access token creations, and app deployment across regions. It is managed by Akka and operates as a globally accessible service.

The Control Plane is designed as a distributed and highly resilient application. In the unlikely event that the Control Plane becomes unavailable, applications running in the Application Plane will continue to operate without interruption, ensuring resilience and availability.

Akka Application Plane

The application plane runs your workloads. It consists of one or more federated regions that run independently of the control plane and each other.

Every region within the application plane is responsible for pulling the application images, deploying it, scaling your application's instances within a region, and connecting the application to its peers running within other Akka regions.

The application plane contains a Kubernetes orchestration cluster, the persistence store the application stores its data in, and a local container registry where application images are cached and can be sourced from.

AKKA

Responsibilities

In a Bring Your Own Cloud (BYOC) model, the responsibilities for system management and maintenance are shared between Akka and your team, covering both initial provisioning and ongoing operational tasks.

	You	Akka
Account	Setup dedicated cloud account with non-human privileged roles for deployment, read-only ops for monitoring, and ops for infrastructure management. Specify human user roles and permissions specific to your organization's security and compliance requirements.	Akka remote access verification.
Bootstrapping	Review default networking, compute, and persistence configuration of Akka with the Akka team. Recommend changes to Akka's default configuration to improve app performance, lower RTO / RPO commitments, or optimize costs. Integrate cloud security services for audit logging.	Infrastructure provisioning and validation. Infrastructure configuration. Create IAM roles for your human users.
Maintenance		Updates to Akka control plane and application plane with security patches and upgrades.
Data Encryption	Establish secrets provided through a Key Management Store.	Rotation of secrets and keys within the control plane and your application planes. Create a root certificate for your region group and intermediate certificates for your regions.
Monitoring	Export application metrics, logs and traces to your observability tools.	Monitoring and availability of the Akka control plane and every application plane.
Backup and Recovery		Infrastructure and persistence backups.
Your Akka App Elasticity		Dynamically add / remove additional compute and persistence to accommodate real-time traffic against your performance SLA targets.

BYOC Installation Process

1. Connect with an Akka solution architect to fill out a short environment and requirements questionnaire.
2. You will create a new account within your cloud provider of choice.
3. The Akka team will provide you with a set of IAM permissions to be applied by you to the VPC environment. This will grant Akka's provisioning and monitoring system access to the account to execute the BYOC provisioning process.
4. Akka will execute the BYOC provisioning process and smoke test the environment. Your region will then appear as a deployment target within your Akka organization and account.

Prerequisites

To deploy Akka regions, our control plane identity requires access to a dedicated cloud account within your VPC to provision region-specific resources. Customers are responsible for granting this access to the control plane identity within their accounts.

You have the flexibility to organize your Akka regions making up the Akka Application plane in multiple ways—either within a single account or across several accounts.

Initial Setup and Configuration

Our GitOps process uses Crossplane and Flux to maintain a declarative approach to infrastructure management. These tools monitor for configuration drift, enabling identification and remediation of deviations. Role-based access controls (RBAC) and audit logging further ensure compliance by preventing unauthorized changes.

As an example, the following outlines how to initialize an Akka region in AWS. To begin, a non-human privileged role must be created. This role is configured with the minimum permissions required to set up and manage the Akka environment. It includes a trust policy that grants the control plane identity permission to assume the role, enabling the control plane to securely bootstrap and manage the environment.

The privileged role in the AWS account will handle the initial infrastructure bootstrapping, release deployments, and ongoing product maintenance. Additionally, it will create other IAM roles for human users based on specific use cases. These roles and permissions are configured with limited time-based access and can be tailored to meet your organization's security and compliance requirements.

IAM Role	Infrastructure and Kubernetes Permissions
Deployment Service Account Role	This IAM role will be created by you and shared with us to provision and manage the account. This role is for our deployment tools. The responsibilities will include Initial setup and configuration of your region and the ability to perform continuous releases via the CI/CD pipeline. Full Create, Read, Update, and Destroy permissions on the following resources: EC2, EKS, IAM, KMS, Logs, RDS, Route53, S3, SecretsManager, and STS. Kubernetes permissions (ClusterRole/ClusterRoleBinding): `cluster-admin/system:masters`
Read-Only Operations Role	This IAM role will be created by us and provides the Akka operations team with read-only access to infrastructure and Kubernetes for monitoring, alerting, and auditing purposes. This role is used most often for administering the product. Read-only permissions on the following resources: EC2, EKS, IAM, KMS, Logs, RDS, Route53, S3, STS Kubernetes permissions (ClusterRole/ClusterRoleBinding): `view/viewer`
Elevated Operations Role	This IAM role will be created by us and allows the Akka operations team to read and update infrastructure and manage Kubernetes. This role will only be used when necessary and the read-only role didn't have enough permissions. Create, Read, and Update permissions on the following resources: EC2, EKS, IAM, KMS, Logs, RDS, Route53, S3, and STS. Kubernetes permissions (ClusterRole/ClusterRoleBinding): `edit/editor`
Limited Administrator Role	This ISM role will be created by us and would provide temporary access escalated from the Akka operations team. This role would only be used in emergency situations such as an outage, degradation, etc. This role is not a full admin, and only has administrative control over the resources we maintain for the product. These permissions are the same as the Deployment Service Account Role. Full Create, Read, Update, and Destroy permissions on the following resources: EC2, EKS, IAM, KMS, Logs, RDS, Route53, S3, SecretsManager, and STS. Kubernetes permissions (ClusterRole/ClusterRoleBinding): `cluster-admin/system:masters`

Configuration

Network Configuration

The network configuration includes a VPC and three subnets per region. Load balancers are also provisioned. In the default configuration, connectivity between the Akka region and the

customer's environment would traverse the Internet. Optionally, customers can use either AWS Privatelink or AWS Transit Gateway to limit exposure on the Internet for communication.

Persistence Store and Compute Sizing

Our default persistence store is db.r6g.2xlarge with 8 vCPU and 64 GB RAM for up to 10K data operations per second.

Our default compute configuration is to use node sizes between m5.2xlarge (8 vCPU/32 GB RAM) and m5.4xlarge (16 vCPU/64 GB RAM) or similar.

Service-Instance Sizing

A deployed Akka application has by default three instances that are distributed over the cloud providers availability zones. Auto scaling can change the number of instances per service based on CPU usage.

BYOC Bootstrap Process

The application plane deployment process consists of two key phases: initial infrastructure provisioning and Application Plane deployment and maintenance. These steps ensure that the required cloud infrastructure is provisioned and that the Application Plane—consisting of one or more regions—is deployed and maintained within your cloud account.

Initial Infrastructure Provisioning

Role Setup

The Akka control plane is granted permissions to assume the Deployment Service Account role in your cloud account. This role allows secure management of resources in your environment.

Infrastructure Creation

The Akka control plane provisions the necessary infrastructure for the Application Plane in the specified regions of your cloud account.

These resources (e.g., compute, storage, and networking) are validated to meet the requirements for hosting the Application Plane.

Application Plane Deployment and Maintenance

Deployment to Regions

After provisioning, the control plane deploys the Application Plane to the configured regions in your cloud account.

The deployed Application Plane regions are then connected to your Akka Organization, enabling centralized management and monitoring.

Ongoing Maintenance

The Akka control plane ensures the Application Plane remains up-to-date across all regions with the latest software updates, including new features, bug fixes, and security patches.

Regular monitoring and validation are performed to maintain the health of the infrastructure and the Application Plane.

Communications

Communication between Akka's Control Plane and Application Planes, as well as inter-region communication within the Application Planes, is designed with security, reliability, and transparency in mind.

Command and Control

Command and control communication between the Control Plane and Application Plane is facilitated through APIs secured by TLS and token-based authentication.

The Akka Control and Application Plane expose APIs: the Management API and Execution API, respectively. These APIs are internet-facing and secured using TLS certificates provisioned by a globally trusted Certificate Authority (CA). All communication with these APIs requires bearer tokens passed in the HTTP Authorization header for authentication.

Inter-Region Communication Within the Application Plane

Applications running across multiple Akka regions in the Application Plane communicate securely using mutual TLS (mTLS). Each client and service within the Application Plane is issued unique certificates which can be issued by Akka's or your KMS, and communication is only established if both sides authenticate each other using these certificates. If authentication fails, communication is blocked.

This transparent and automated mechanism, managed by Akka, relies on a multi-region ingress service embedded in each region. It ensures seamless communication and eliminates the need for manual configuration

Security and Compliance

Akka adheres to a wide range of compliance standards to ensure the security and integrity of our products. Comprehensive information on the company's compliance and certifications is available in the Akka trust center.

Data Protection and Encryption

All application data is stored within an encrypted, durable event store which cannot be externally queried. Application data at rest is encrypted using the store's default mechanism. Encryption keys can be provided by Akka or through your Key Management Store.

Additionally, we offer a further layer of encryption for:

- Secrets that are synced to regional execution clusters from the control plane through the management-api.
- Secrets associated with authentication, such as TOTP secrets and OpenID refresh tokens.
- Secrets provided by the customer through environment variables or your KMS.

We apply regular and automated key rotations for data at rest, using Data Encryption Keys (DEK) and Key Encryption Keys (KEK).

DEKs encrypt data and are stored with it, encrypted by a KEK. KEKs are used to encrypt DEKs. KEKs are stored separately as Kubernetes secrets and are rotated periodically.

When a KEK is rotated, all associated DEKs are re-encrypted without re-encrypting the underlying data, ensuring efficient and seamless re-encryption.

Monitoring and Management

Akka uses Groundcover (groundcover.com) internally for monitoring, observability, and alerting for infrastructure and core platform services. Customers must configure Observability and monitoring by exporting application metrics, logs, and traces to their monitoring tools. Akka is responsible for the control plane including all platform services and you are responsible for monitoring the health of your own applications.

Akka manages the underlying compute nodes and database sizing. Compute nodes are elastic and will grow and shrink with the number of deployed services. The database storage autoscales but additional CPU and memory for the database is an operation that our team applies based upon your applications exceeding utilization thresholds caused by repeated traffic spikes.

On request, Akka can pre-warm environments to provide additional capacity in anticipation of sudden utilization spikes.

Logging and Audit Trails

Akka uses various security configurations in our own cloud environments. You can enable cloud security services like GuardDuty, Config, and CloudTrail with a custom configuration decoupled from Akka's provisioning process.

Infrastructure Resource Provisioning

Infrastructure resources within Akka Application Planes are managed to ensure optimal performance and scalability. The provisioning and scaling of compute nodes and database resources are detailed below.

Compute Nodes

Akka Services are scaled horizontally between a minimum and maximum instance count based on CPU usage. This scaling activity determines the demand for compute nodes in the region.

The number of compute nodes can scale dynamically to meet the workload demands of the region, ensuring sufficient resources for service instances.

The maximum number of compute nodes is configured based on the expected workload and the number of service instances required for the region.

If resource utilization approaches this configured maximum, the Akka team will increase the limit and notify the customer to ensure continued performance without disruption.

Database Resources

Database storage automatically scales to accommodate increases in data volume without requiring manual intervention.

If additional database compute resources (e.g., CPU or memory) are needed beyond the initial configuration, the Akka team can resize the database to handle the increased workload.

This approach ensures the infrastructure remains flexible and responsive to workload changes while maintaining high availability. Customers will be informed of any adjustments that impact service.

Backups and Disaster Recovery

Kubernetes resources and databases are backed up and recoverable. Persistence RPO is 5 minutes and RTO is 24 hours. Kubernetes RTO / RPO are 1 hour. Infrastructure upgrades can lower RPO / RTO numbers. Infrastructure and configuration upgrades can improve these commitments.

Akka provides data exporting capabilities that allow exporting of your data to external storage for additional processing or long term storage.

Support and Maintenance

The Akka team continuously applies security patches and software upgrades. Maintenance includes infrastructure upgrades and modifications necessary for the platform to remain secure and performant.

If any issues are identified, we have a customer support portal where issues can be created according to our Customer Support Policy. Issues are worked through based on severity and impact.