International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P109
Eureka Vision Publication | Volume 4, Issue 2, 78-100, 2023

Original Article

Multi-Region Contact Center Resiliency with Amazon Connect

Naveen Narayan[1], Prashanth Krishnamurthy[2], Ramprasad Srirama[3]
[1]Sr Solutions Architect, Amazon Web Services, USA.
[2,3]Sr Partner Solutions Architecture, Amazon Web Services, USA.

Abstract - In a digital age of international service provision, organizations are required to guarantee unbroken customer service experiences. Single-region contact center implementation can lead to infrastructure outages and outages in regions or maintenance windows that present a danger to service delivery and customer satisfaction. Amazon Web Services (AWS) releases Amazon Connect Global Resiliency with an impressive number of APIs and architectural tools, which automate and provide options to deploy contact centers across regions in a multi-regional way. This paper provides a detailed description of this solution that enables organizations to duplicate their Amazon Connect across regions and programmatically control traffic using Traffic Distribution Groups (TDGs). The given sample user interface, registered through CloudFormation, S3, and CloudFront, facilitates working with resiliency APIs and provides immediate visual feedback through structured JSON outputs. We dive into its deployment lifecycle, including initial configuration and AWS Identity and Access Management (IAM) permissions, and cover the management of telephony functionality, which includes claiming phone numbers, replication, and cross-region phone routing. Use cases in real life, such as automated failover, traffic splitting every 10%, and disaster recovery testing, are examined. Moreover, we discuss the performance features, pricing issues, security concerns and limitations of system onboarding. The purpose of this paper is to instruct practitioners on how to develop highly available contact centers with cloud-native practices in order to provide continuous support to global customers.

Keywords - Amazon Connect, Global Resiliency, Disaster Recovery, AWS CloudFormation, Traffic Distribution Group (TDG), High Availability.

1. Introduction

Customer engagement is more important to businesses in this age than ever before, and as such, there is a higher demand for the continuous availability of call centers. Businesses need to guarantee that their support systems are available at any time, whether or not there is a regional outage or some unforeseen interference. [1-3] Trusting exclusively in single-region deployments typically subjects the contact center to the chances of downtime due to reasons like infrastructure failure, natural calamities or localized network problems. Organizations are therefore developing multi-region, cloud-native architectures that provide fault tolerance, geographic resilience and redundancy to overcome these challenges.

The Amazon Connect Global Resiliency helps fulfil these requirements by allowing organizations to have multiple contact center instances across Amazon Web Services (AWS) regions. Administrators can replicate Amazon Connect environments using a set of APIs, along with a related web-based interface, to traverse Traffic Distribution Groups (TDGs) and control regional call routing, as well as invoke failover mechanisms. [1-3] The current paper discusses the end-to-end solution to the development of such resilient architectures with Amazon Connect; specifically, the discussion of key design principles, deployment automation techniques, security measures, and managing costs. This is aimed at enabling IT and cloud specialists to provide a stable global approach to a contact center experience, without losing compliance and operating efficiency.

2. Related Work

High availability in traditional contact centers was attained by utilizing redundant on-premise infrastructure and rigid routing schemes, typically utilizing costly and complex hardware stacks, PBXs, SBCs, and load balancers. These arrangements were highly dependent on manual assistance and disaster recovery mockups and lacked flexibility in the event of outages or sudden

high-demand situations. Failover processes generally comprised replicating whole constructions in variously located data centers, which raised management fees and presented wait or synchronization issues. In addition, legacy architectures were not agile and therefore could not conform to current expectations of real-time scaling, self-healing systems, and easy cross-region disaster recovery.

Amazon Connect, on the other hand, offers a cloud-native, contemporary environment for contact center operation. Being a fully managed service with consumption-based pricing, it also eliminates the need for infrastructure and offers all the robust features, including omnichannel engagement, intelligent routing, and AI integrations. Although it was initially designed to provide regional-scale resiliency by the deployment of multiple AZs, it did not support cross-region failover, which was changed with the launch of Amazon Connect Global Resiliency. It scales Amazon Connect to best practices beyond contact centers into industry-wide cloud disaster recovery by allowing administrators to programmatically clone contact center instances and dynamically distribute contact center traffic among regions with Traffic Distribution Groups (TDGs). The Global Resiliency of Amazon Connect provides more granular control, automation, and a more solid basis for global, enterprise-level high availability compared to what other providers, such as Twilio Flex or Genesys Cloud, may offer with PoPs or static failover.

3. Amazon Connect Global Resiliency Overview

3.1. Motivation behind Global Resiliency

The diagram shows the fundamental deployment flow and architecture of the Global Resiliency interface of Amazon Connect. The initiation of the process occurs when a predefined infrastructure stack is deployed by an administrator or an API tester using AWS CloudFormation. [4-7] This template coordinates the setup of cloud services like an Amazon S3 bucket to store web materials and the CloudFront distribution to deliver the User Interface (UI). The UI enables the control of resiliency tasks, such as instance replication, traffic redirection, and phone number handling, across AWS regions. It is a core building block of the new resiliency pattern with Amazon Connect: all configurations are infrastructure-as-code based to be reproducible and automated.

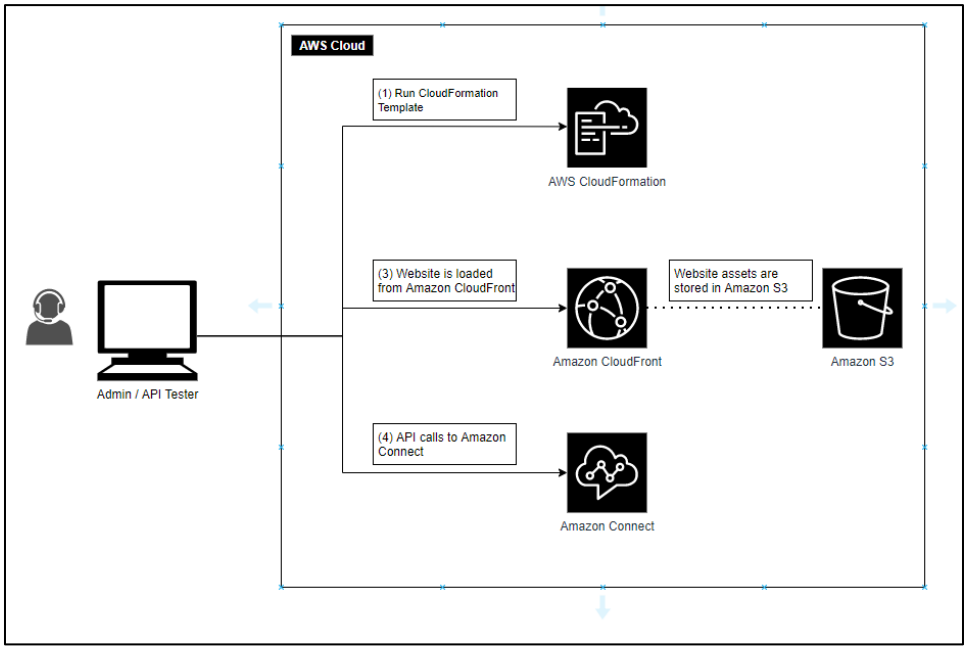


Figure 1. Overview of Amazon Connect Global Resiliency

After deployment, the admin uses the URL provided by Amazon CloudFront to access the web-based interface. The CloudFront edge server network will transmit the web UI data directly to the Amazon S3 bucket, providing global scalability and quick access to web content due to its low latency. This interface enables the admin to invoke API actions, such as creating a replica Amazon Connect instance in another region, creating Traffic Distribution Groups (TDGs), and managing phone numbers. The UI makes API calls that are securely forwarded to Amazon Connect, where the related control plane runs and executes the

specified operations. The flow presented in the diagram separates the concerns: content delivery, data storage, and service invocations are modularized on the basis of the core AWS components.

Fault tolerance, automation, and ease of deployment are some of the foci of this architecture. Administrators can use this fully managed solution to enable multi-region failover capability within minutes, without manual configuration or operation of disaster recovery mechanisms. The system also fosters the adoption of best practices in cloud infrastructure deployment and contact centre resilience, as it utilises CloudFormation to orchestrate and a lightweight web front as an interaction medium. The diagram symbolically depicts not only the interaction path but also a cloud-centered way of enabling global resiliency without an operational overhead.

3.2. API Capabilities and Use Cases

Amazon Connect Global Resiliency is a rich framework of APIs that gives an organization the capability to create, manage and monitor the multi-region contact center deployment in a way never done before. These APIs enable administrators to mirror Amazon Connect devices across two AWS Regions, configure Traffic Distribution Groups (TDGs) to manipulate telephony routing, and dynamically switch phone numbers between instances or TDGs. The distribution and placement of traffic through the primary and secondary regions can be programmatically controlled by administrators. The system offers the flexibility to adjust traffic in 10% increments, supporting both gradual and instantaneous failover strategies. For example, a business may direct 90% of traffic to the main area and allocate 10% to a backup region for testing purposes. Traffic may be completely redirected to the secondary region in minutes, if necessary. All of this is achieved through a simple API call. Other possible applications involve disaster recovery tests, region-based performance optimization, and data residency-based routing based on compliance requirements. All of these capabilities are accessible to you through the AWS SDK and CLI, but also through a visual interface that uses APIs on top of these APIs, so that operations teams can manage them either code-driven or through a UI.

3.3. Supported Regions and Limitations

Amazon Connect Global Resiliency is currently limited to region pair deployments, with US-East-1 (N. Virginia) and US-West-2 (Oregon) as the primary regions allowed to replicate each other. This drawback ensures that AWS can deliver a consistent API performance, sync, and routing path between the two regions. Every instance launched as a part of a Global Resiliency configuration should be SAML-configured and should be a member of an allow-listed AWS account. Additionally, the cases have to be issued in a paired region combination where AWS directly supports the cases, after which organizations have to collaborate with the AWS account manager to obtain access. These restrictions are implemented to ensure consistency in their operations and prevent configuration mistakes in region pairs. Although this initially constrains global coverage, the design establishes a solid foundation that can be extended in the future to cover other regions, as well as facilitate cross-continental failover.

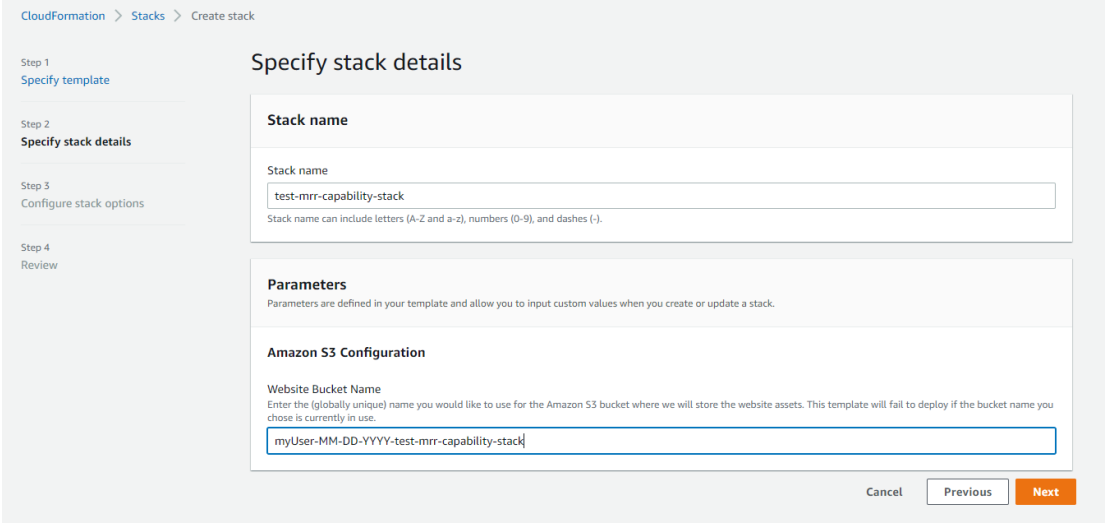


Figure 2. CloudFormation Stack creation screen

CloudFormation stack setup screen where the administrators specify all the deployment parameters required to activate Global Resiliency. This picture shows the insertion of a globally unique stack name and Amazon S3 bucket name to which the static assets required to render the user interface will be uploaded. This screen plays a pivotal role in the provisioning process, which initiates all the AWS resources required to provision Amazon Connect API management resources across regions. When implemented, this configuration can provide the distribution of traffic using TDGs- logical entities where it is possible to dynamically control telephony call routing between a primary and secondary region. The form is more representative of the start of a deployment; it is the start of a resilient and globally aware customer service infrastructure.

3.4. Access and Onboarding Requirements

Amazon Connect Global Resiliency is now gated, and the feature induces explicit onboarding. To enable this capability, organizations should directly coordinate this with their AWS Account Manager to have their accounts allow-listed. This ensures that only environments that are ready and compliant utilise the APIs in the production of their workloads. Among them is that the Amazon Connect we currently have should be SAML-enabled, allowing secure single sign-on and integration with federated identities. This is especially critical since replication between zones is only possible due to the assumption that identity and access patterns are consistent across occurrences. Also, an existing instance must already have at least one or more phone numbers claimed, which will be unmapped (or, in the case of Traffic Distribution Groups (TDGs)) claimed during the resiliency configuration. It is important to mention that Global Resiliency is currently limited to U.S.-based AWS Regions, specifically us-east-1 and us-west-2, so it is only available to organizations whose operations are in or are targeting these jurisdictions. This limitation, however, is expected to change, as AWS is anticipated to offer more support for global telephony routing and region-to-region replication.

4. Architectural Framework

4.1. High-Level Architecture and Deployment Strategy

Amazon Connect Resiliency with Global deployment is deployed using a multi-region approach where the core instance of Amazon Connect is replicated into a partner Region of AWS. [8-11] The system is implemented based on the idea of controlled redundancy, where the main instance serves the major part of the traffic. In contrast, the secondary instance (replica) is always in a ready or active state, so it can take over when the primary instance fails. The architecture employs Traffic Distribution Groups (TDGs) as the voice control plane used in directing voice traffic between any two regions. This logical grouping enables fine-grained control of traffic, allowing administrators to dynamically balance the traffic load of telephony calls according to failover provisions, performance tuning, or standard testing. This architecture can be managed through a secure web-based UI or API endpoints by the administrators and assists in seamless control across regions without any traditional network failover scripts or third-party telephony switching hardware.

4.2. Role of Amazon CloudFormation, S3, and CloudFront

Amazon Connect's Global Resiliency deployment and management experience is based on a serverless, infrastructure-as-code approach, with a key focus on AWS CloudFormation, Amazon S3, and Amazon CloudFront. A CloudFormation script that provisions all necessary resources is utilized, such as the S3 bucket and CloudFront distribution and maximizes consistency and repeatability when it comes to deployment. The S3 bucket contains the assets of the static website, which is the frontend of the administration web application. Amazon CloudFront plays the role of the content delivery level, the content delivery network that can provide low-latency to the interface served in a geographical location array, offloading the origin. This architectural model enables the separation of deployment logic, UI rendering, and API execution into modular AWS services, providing a scalable, fault-tolerant, and cost-effective solution. It also enables organizations to custom-implement the interface rapidly in any region in a couple of minutes without any manual configuration, consistent with the idea of cloud-native DevOps.

4.3. Identity and Access Management (IAM) Policies

Fine-grained IAM policies are applied to provide security and access control based on the principle of least privilege. Administrators are required to set up roles that enable specific activities related to Amazon Connect, S3, CloudFormation, and TDG APIs. The CloudFormation template used with the Global Resiliency solution has IAM roles and permissions configured to

facilitate the successful setup. The sample template provides minimally scoped policies to showcase how to build up the solution, but production environments ought to be configured differently so that IAM roles suit organizational requirements of security and compliance. Notably, API calls made via the UI must be executed with credentials that have adequate permission scopes, and secrets (such as access keys) must be handled safely through environment-specific settings or by integrating with AWS Secrets Manager. It is essential to ensure that IAM policies are appropriately scoped and audited to minimise exposure, particularly in cases where resources are distributed across multiple AWS Regions.

5. Implementation Methodology

5.1. Solution Setup Using CloudFormation

This Amazon Connect Global Resiliency deployment begins with the provisioning of the required infrastructure, which is partially achieved by utilising AWS CloudFormation. AWS has a ready-made CloudFormation template, which contains a definition of all the resources required to deploy the management interface. [12-16] This involves creating an Amazon S3 bucket to host web assets and a CloudFront distribution to distribute the user interface to administrators. When creating a CloudFormation stack, administrators must provide a globally unique name for the S3 bucket and the name of the deployment stack. After creating the stack and bringing it to a state of CREATE_COMPLETE, the UI can be accessed through the generated CloudFront URL. This deployment approach is also in line with the latest infrastructure-as-code (IaC) frameworks, which enable firms to automate and replicate the arrangement between environments. It also provides standard configuration and policy enforcement, minimizing the chances of misconfiguration. Notably, the CloudFormation template was configured with IAM roles and policies that provide the least-privileged access to secure and restricted AWS resources, including Amazon Connect, S3, and CloudFront.

This picture demonstrates the last output segment of the CloudFormation stack deployment. It marks the URL of CloudFrontEndpoint, which is the service gateway used for Global Resiliency operations administration user interface. The CloudFront endpoint is generated on demand and provided as an output of the stack deployment completion, appearing as the endpoint for all interactive administration. The configuration identity to run the Global resiliency stack. Required fields, such as the name of the unique stack and the S3 bucket identifier, are also displayed. The administrator also receives a prompt that recognizes the creation of IAM resources, because the template creates the roles required. This interface plays a crucial role in creating reproducible, regionally scoped infrastructure with minimal manual procedures.

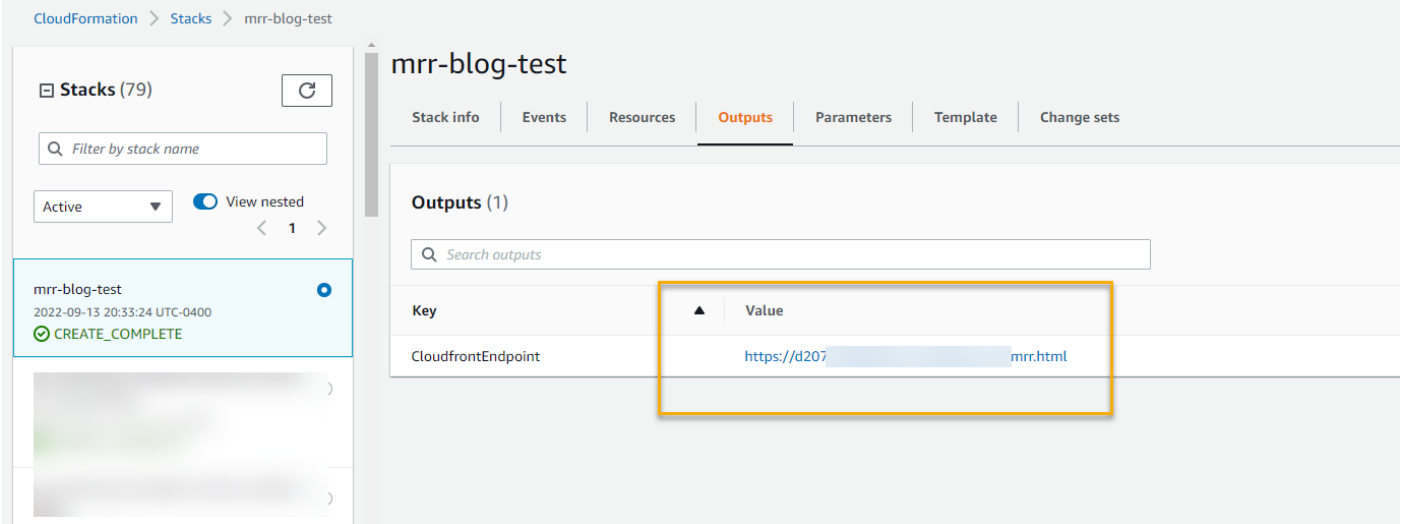


Figure 3. CloudFormation Outputs tab showing CloudFront endpoint after stack deployment

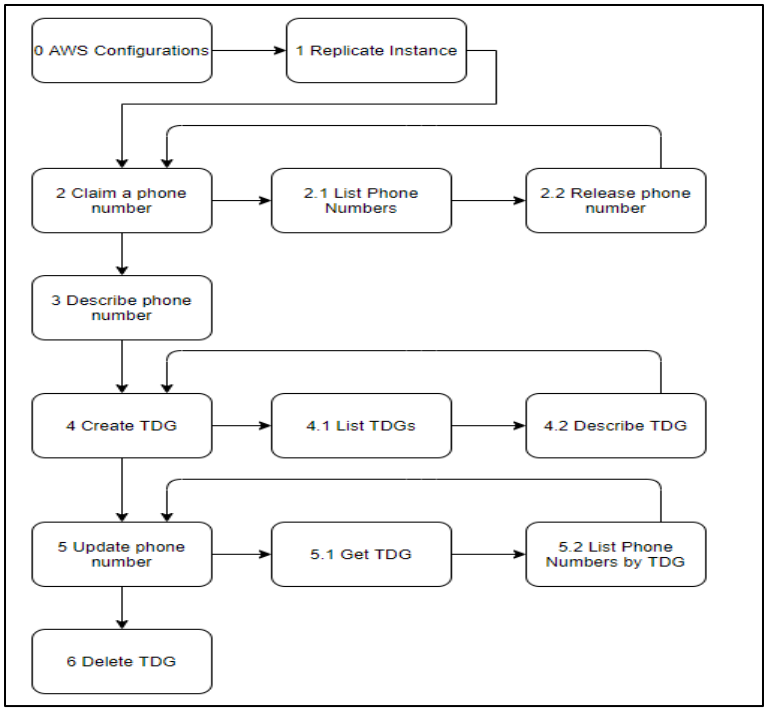


Figure 4. Workflow diagram showing API operation sequence for global resiliency

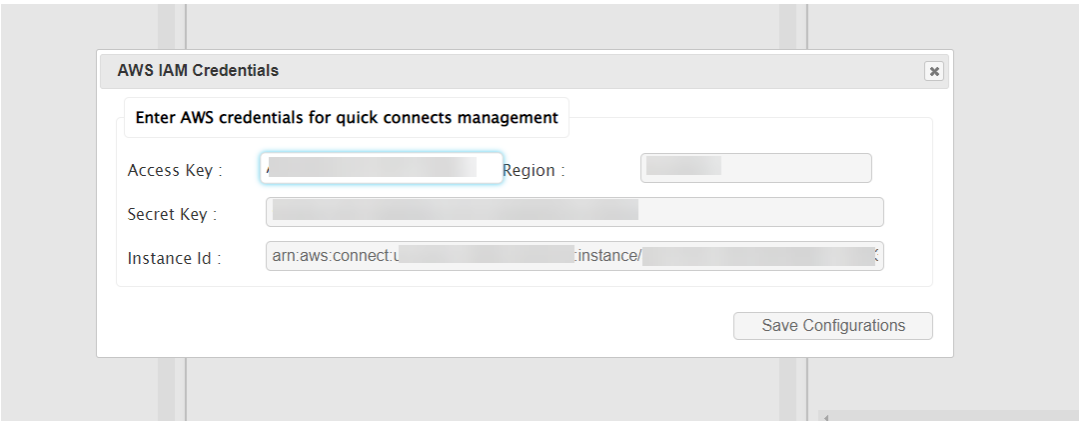


Figure 5. AWS Config input form (Instance ID, Region, Access Keys)

5.2. Deployment in Primary and Replica Regions

The use of multiple regions in Amazon Connect, with primary and replica regions, undermines the essence of robust contact centre architecture. This is initiated by provisioning a production-ready instance in the main AWS region and then using the backend administrative UI, hosted over CloudFront, to create a copy of the same instance in the supported secondary region through the instance API, 'replicate instance'. This replication enables the copying of configurations, such as routing profiles, queues, and prompts, which are replicated correctly. When the two cases are logically connected, predetermined call traffic can be distributed among them using Traffic Distribution Groups (TDGs), with values ranging from 10 to 100, in increments of 10% of the total traffic. The interface enables the claiming or mapping of phone numbers to TDGs, thus providing telephony continuity. This configuration reduces the possibility of downtime because its replica instance can be turned on within a moment in case the major region experiences dysfunctions, providing business contingency and a better customer experience.

5.3. Walkthrough of the Web-Based Admin UI

It features an administrative UI accessible via the web, with an emphasis on ease of operation and efficiency, along with an intuitive four-column interface known as the UI Pillars. Pillar A contains API control buttons that enable several important actions, including instance replication, TDG creation, and phone number management, with just a single click, thereby concealing the complexity of direct AWS API or CLI use. B Pillar is a list of available TDG and also enables contextual operations such as distributing traffic or displaying status. Pillar C deals with the management of phone numbers, allowing users to view, update, or release numbers associated with a particular TDG or instance. Pillar D is a live feedback console where API calls that have been executed display structured JSON responses, allowing them to be reviewed for debugging purposes. Such a structured setup enables technical groups as well as business administrators to operate and observe multi-region resiliency settings without requiring them to write code or comprehend the complexities of the underlying cloud infrastructure.

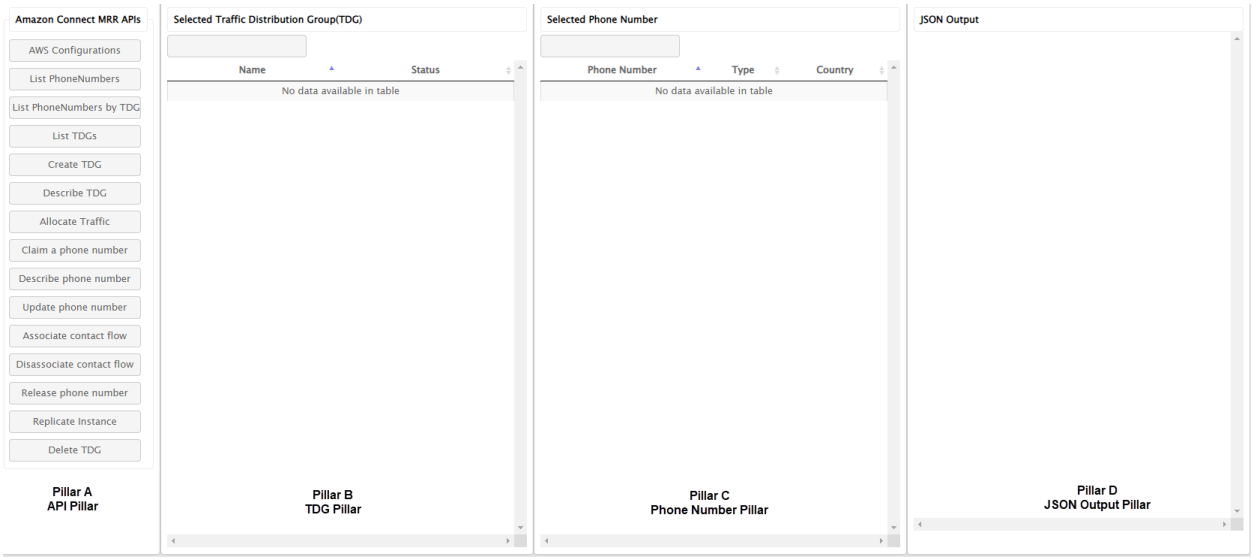


Figure 6. UI Pillars Layout Showing Pillars A–D

5.4. AWS Configurations & Instance Linking

Amazon Connect Global Resiliency features require configuration first on the AWS environment through the submission of required credentials and instance metadata. Accessing this process requires the user to log in to the web-based UI using the AWS Configurations group in Pillar A. In this case, they present an AWS access key and a secret key, and then specify the Amazon Connect instance ID and the region where it is located. This input is essential for communicating with the administrative interface and selecting the correct Connect instance to make API requests. When properly set up, this connection enables all the other pillars (TDG management, phone number processing, and JSON logging) to function in context. This is an effective action that makes the interface strictly connected with the instance of Connect in the primary region and makes the environment ready to carry out the operations of replication and traffic control among paired regions.

An Amazon Connect Global Resiliency user interface is organized in the form of a four-pillar layout that can be understood to symbolize the functional parts of contact center management across regions. Pillar A, a command center of the UI, holds all API buttons that a user can take actions such as Create TDG, Replicate Instance, and Claim Phone Number. These buttons allow global administrators to easily perform all the necessary actions of AWS APIs, reducing Amazon Web Services complexity, thus not requiring the administrator to code or to use the AWS Console. This centralized design will make exposure to functionality more streamlined and facilitate efficiency, minimize chances of operational errors when configuring or failover operations. Pillars B, C, and D provide dynamic context and feedback, depending on the actions taken in Pillar A. Pillar B handles TDG control, displaying all configured Traffic Distribution Groups and offering access to various operations associated with them, such as traffic allocation or description of the TDG. Pillar C is the telephone number control panel, which displays every DID and TFN number associated with the current setting, including routing and metadata information.

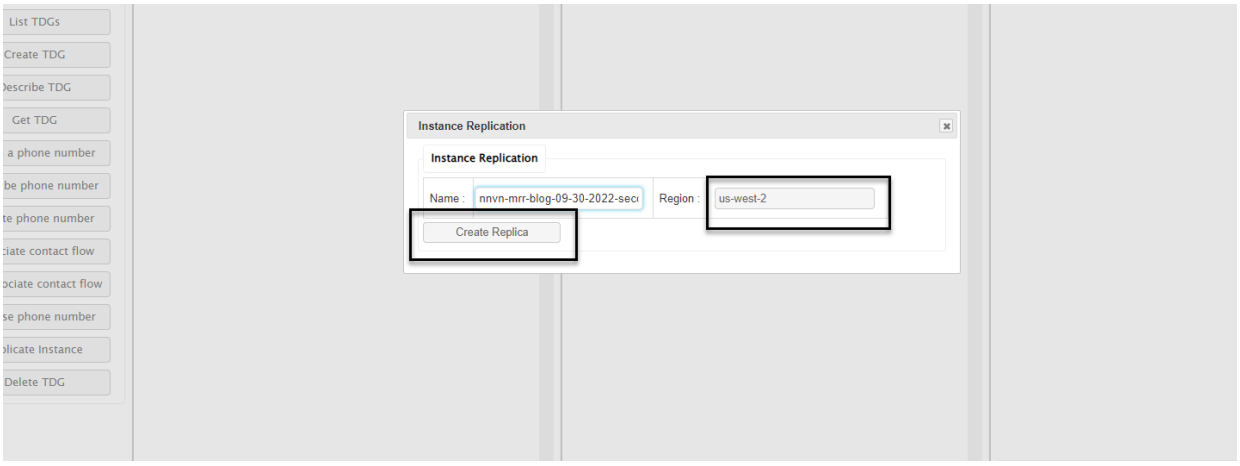


Figure 7. "Replicate Instance" action button

Figure 8. JSON output from Replicate API

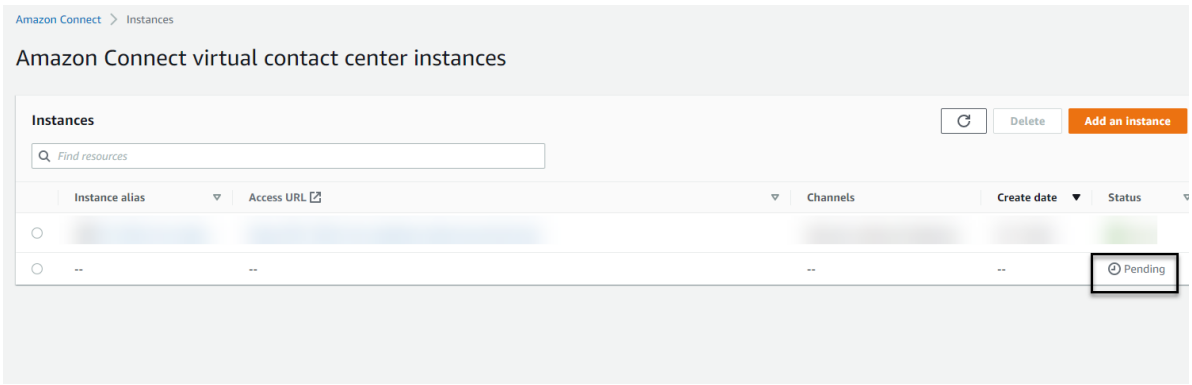


Figure 9. AWS Config input form (Instance ID, Region and Access Keys)

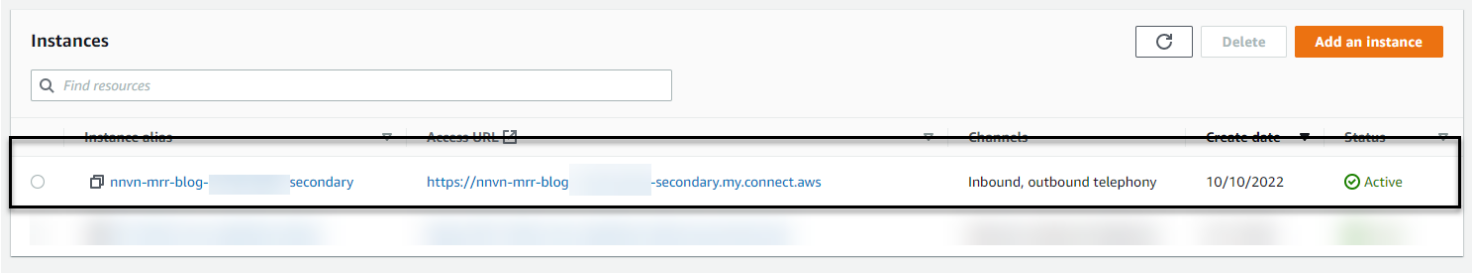


Figure 10. Amazon Connect Instance Page Showing Active Replica

5.5. AWS Configurations & Instance Linking

The AWS Configurations module serves as the starting point for establishing a safe and consistent connection with the administrative UI and Amazon Connect services. In this configuration, the administrator will be required to enter four key parameters: the AWS Access Key, Secret Key, Connect Instance ID, and the associated AWS Region. This input data manages an authenticated session to scope all future API calls to a particular environment, thereby ensuring that, e.g., instance replication or phone number management is performed with security and context. Temporary storage of these credentials in the session maintains security and enables interactive functionality. The step, nonetheless, is quite sensitive to precision; any miscalculation of credentials or region setup may disrupt API communication, which underscores the importance of assigning proper IAM permissions and subsequent validation during the initial configuration setup process.

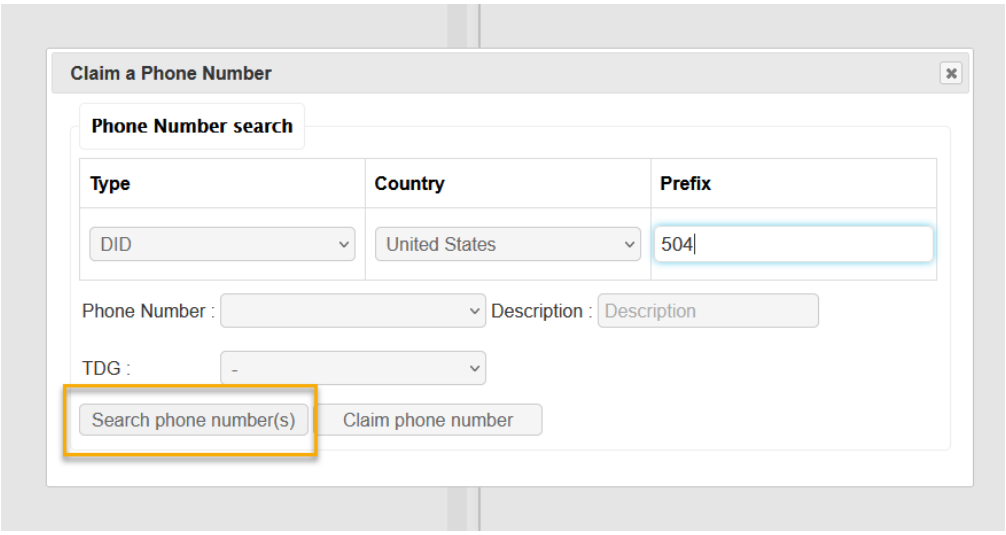


Figure 11. Claim Phone Number Form – Input Fields

The configuration input screen used when making a phone number claim allows administrators to create a secure AWS connection, initiating the process to acquire a Direct Inward Dialling (DID) or Toll-Free Number (TFN). This simplified interface records critical data, including the type of number, target country, and prefix. It optionally links the number to a replaced Traffic Distribution Group (TDG) through a drop-down choice. If no TDG is chosen, the count reverts to the default value set for the Amazon Connect instance being configured. This feature serves as a frontend API to Amazon Connect phone number APIs, providing easy-to-use functionality with no requirement for direct SDK or CLI usage. The incorporated prefix-based search promotes operational efficiency by enabling rapid filtering and acquisition of numbers to meet geographic or functional preferences. After claiming, the number can be used instantly in the contact centre system, making it an asset for handling region-aware traffic steering and telephony redundancy.

6. API Operations and Use Cases

6.1. Creating and Managing Instances

A core API feature introduced by Amazon Connect Global Resiliency is the Replicate Instance functionality, which allows administrators to replicate an existing Amazon Connect instance across AWS regions. Through the web-based UI, such an operation stimulates the replica of a structurally resembling instance, without agents or historical data, in a supported secondary region, like us-west-2 or us-east-1. [17-20] The new instance is rationally connected with the first one under the same Instance Group, which allows managing it centrally and provides the possibility to integrate with TDGs to distribute traffic. Administrators can initiate this activity by providing a distinct identifier for the replica and the target AWS location, which is recommended to use a consistent naming pattern (e.g., instance-secondary). The most common duration of the fulfilment is 5-8 minutes, and the status of the replication is visibly followed through the Amazon Connect console. This API feature is the basis of the fundamental value of Global Resiliency, which is the ability to rapidly replicate in code-free environments, reducing by an order of magnitude the difficulty of deploying multi-region failover strategies.

A graphical image of the Replicate Instance button on Pillar A on the administrative UI. This button triggers the replication API flow, and it is one of the essential functions in the Global Resiliency toolkit. Upon clicking, a dialog box is issued where the user is to provide the new name of the instance and the target replication region. When submitted, this input is packaged into an API call, which in turn provisions the replica Connect instance in the background by UI. This button, whose simplicity masks a considerable degree of architectural complexity, automates the full-stack provisioning of an outcall contact centre, eliminating the need to write a script or manually configure or risk replicating an infrastructure. This is a visual and interactive process that suits the needs of users with little to no experience with AWS CLI or AWS SDK, enabling them to build a reliable, multi-region supporting infrastructure with confidence.

6.2. Phone Number Management Operations

6.2.1. Claiming TFN/DID Numbers

The sequential process of obtaining new telephony numbers using the Claim phone number app in the Global Resiliency admin UI. The administrators start by typing in the available numbers through type, country, and prefix search. The display of search results in a dropdown makes it possible to select the desired number and become its owner with a single click. This will send an API request to Amazon Connect, reserving the number and scaling this up to a particular Traffic Distribution Group (TDG) and connecting that traffic distribution group directly to the Connect instance in the AWS configuration step. The intuitive interface makes the complex process of telephony provisioning fast and easy, enabling enterprises to add sites to their regional presence through local numbers within seconds. The system also responds with a JSON dump, which shows the ARN of the number, region, and current mapping, indicating successful acquisition and immediate availability for application in routing logic.

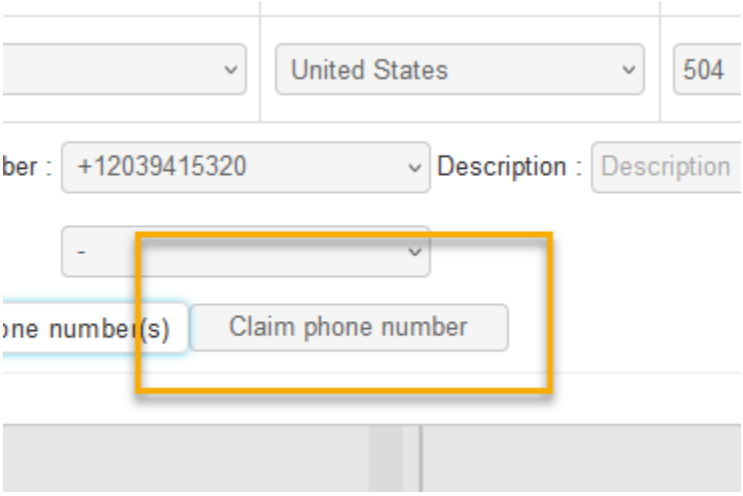


Figure 12. Claim Phone Number Input Form

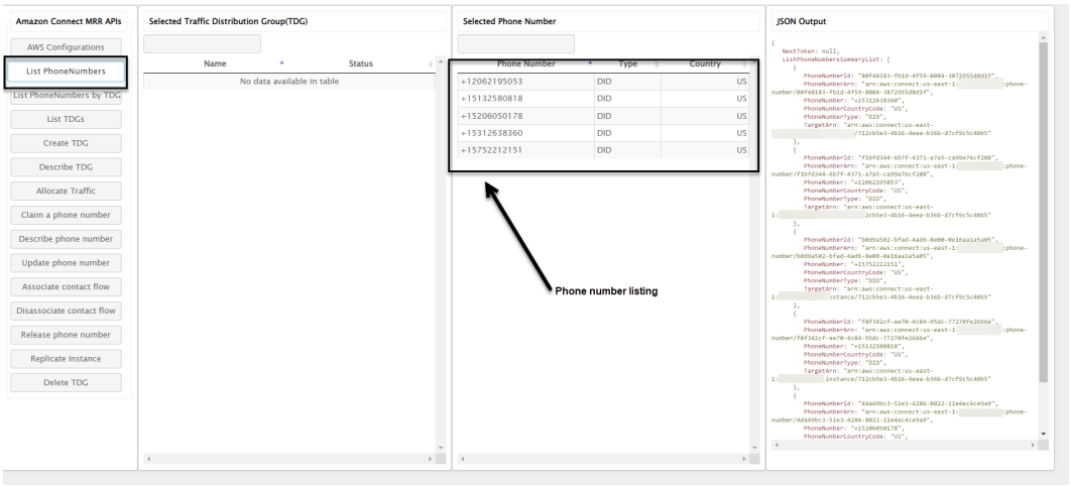


Figure 13. Phone Number List Result from API

6.2.2. Listing and Describing Phone Numbers

Once phone numbers are claimed, administrators must correctly administer and audit their telephony resources across different regions. The Global Resiliency UI comes with such support built in with the operations of List PhoneNumbers and Describe Phone Number. The listing feature lists all the phone numbers that have been attached to the currently selected Amazon Connect instance or TDG, enabling teams to manage the list of their assets that are provisioned. Meanwhile, the describe function provides in-depth metadata for a given number, including Amazon Resource Name (ARN), deployment region, and routing association status. The capabilities are essential in the continued operation of functions that include routing validation, capacity planning and compliance checking. Through these APIs, the contact centre workforce can ensure that every phone number is properly configured and is not idle, thereby eliminating the risks of misrouting and enhancing visibility across failures.

6.2.3. Updating Phone Number Mappings

The List PhoneNumbers acts in the admin UI. After being triggered, the Phone Number Pillar is then filled with a list of the phone numbers together with associated metadata such as type (e.g., DID) and country (e.g., US). At the same time, the JSON Output Pillar represents the unprocessed API result, as phone number IDs, ARNs, and TargetARNs are stored there to specify the Amazon Connect instance linked to the number. This two-view (UI + JSON) approach offers a pleasant, high-level monitoring and technical verification experience. This transparency is precious to run large contact centers that have several dozen or hundreds of numbers in various regions.

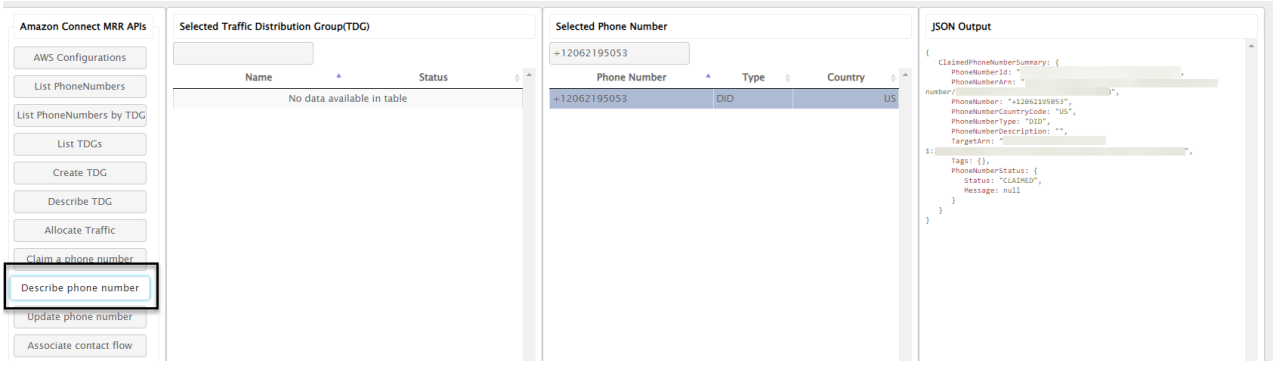


Figure 14. Phone Number List Result from API

The outcome of the Describe phone number procedure. Once you choose a number in the Phone Number Pillar and trigger the activity, the system will provide you with structured JSON information containing some of the defining characteristics of the number. This brings with it the PhoneNumberARN, Type, CountryCode, and above all, the TargetARN, which displays the Amazon Connect instance where the number is currently assigned. A Status object, in addition, validates the status of the number

in terms of being active, e.g. “CLAIMED). This data can assist administrators in verifying the type of call routing, troubleshooting mapping problems, and ensuring that traffic flows are mapped to the proper regions or TDGs in the event of a failover or distribution.

6.2.4. Releasing Numbers

6.2.4.1. Creating Traffic Distribution Groups (TDGs)

Traffic Distribution Groups (TDGs) play a pivotal role in delivering high availability in a world-resilient Amazon Connect solution. They are used as logical routing layers to distribute incoming calls across various instances in the region, where an administrator can control how traffic is balanced, such as through load sharing, disaster recovery, or staged migrations. A TDG can be created through the administrative web UI, where a user clicks the "Create TDG" button, which generates an API request sent to Amazon Connect to create a group in the current AWS environment. The administrators have to give the administrator a unique name and an optional description to aid in determining their use or regional linkage.

After making the API call, the TDG will be presented in the TDG pillar of the interface, in a live status with messages such as CREATION_IN_PROGRESS or ACTIVE, allowing the user to track the progress of the provisioning lifecycle. Once the group is formed, it can be linked to phone numbers and set to provide the percentages of traffic (in 10% increments) between the connected instances. This dynamic configuration enables administrators to redirect traffic on demand and can be used to implement coordinated changes as well as emergency failover conditions. TDGs, therefore, form a basis for establishing a solid, multi-regional contact center that can be shaped around outages and fluctuating demand.

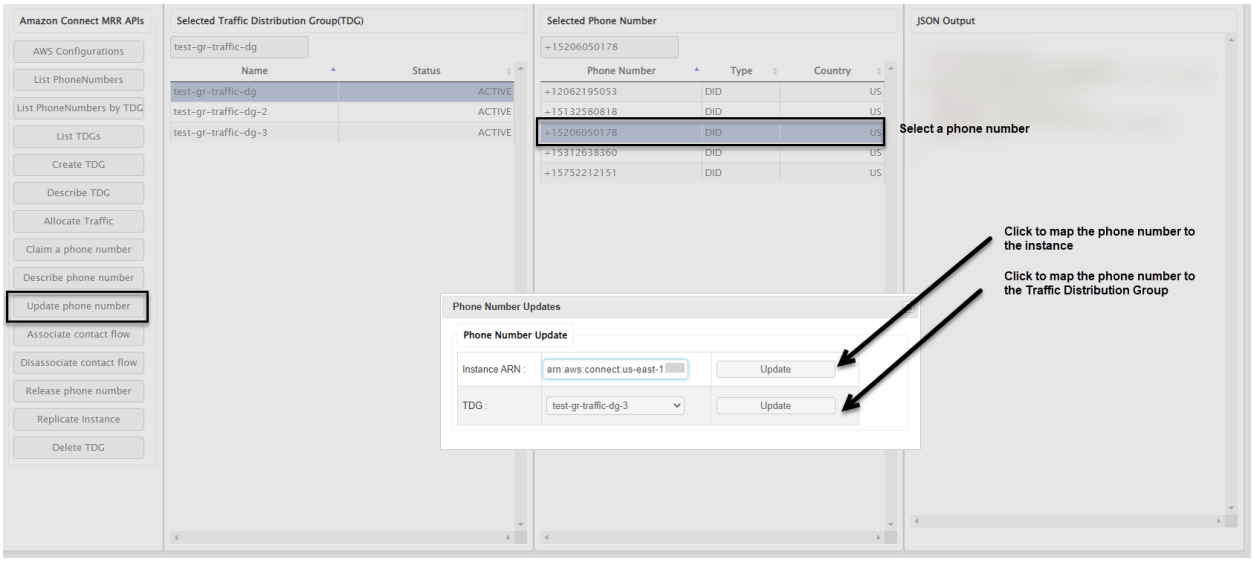
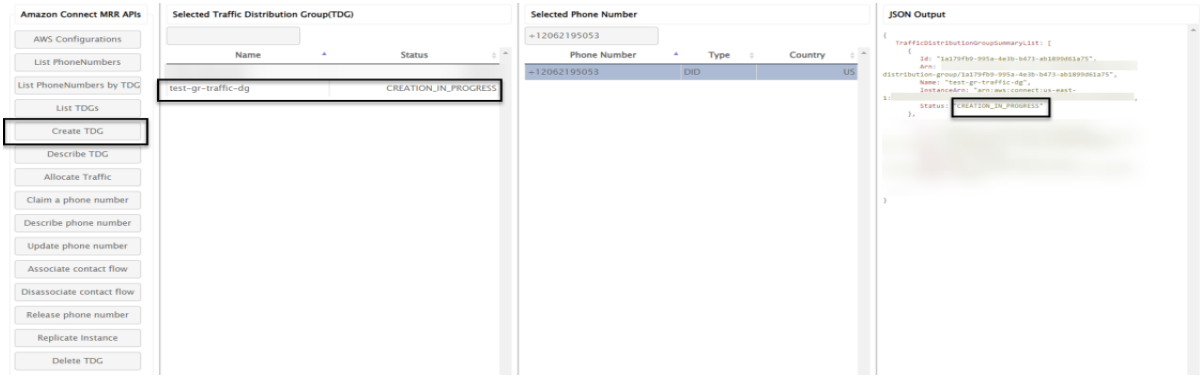


Figure 15. Describe Phone Number Output with Instance ARN



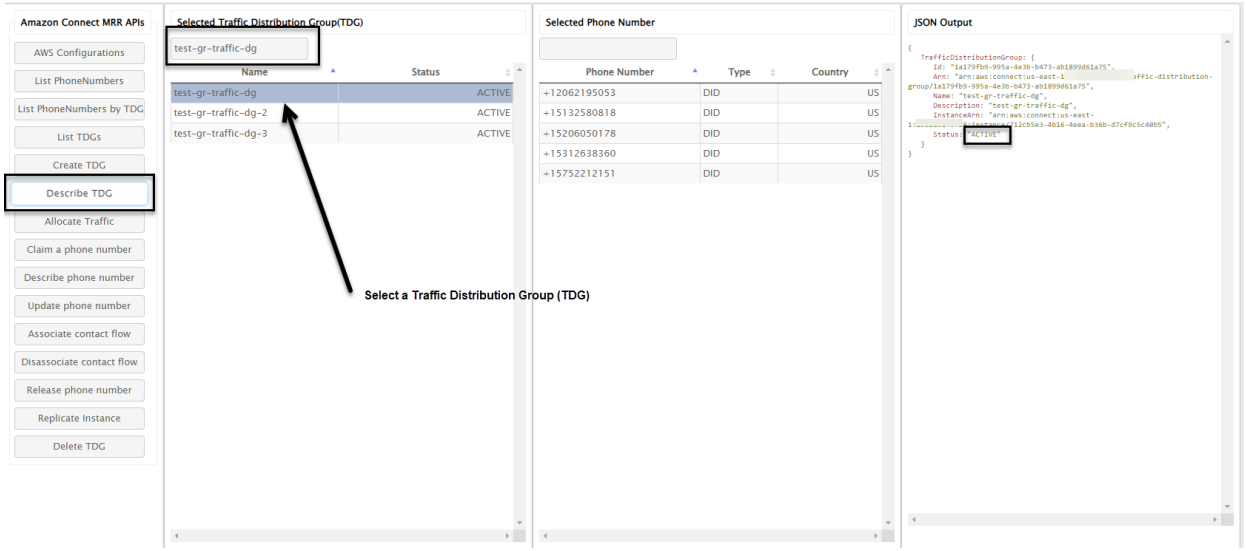


Figure 16. Create TDG Screen in the UI

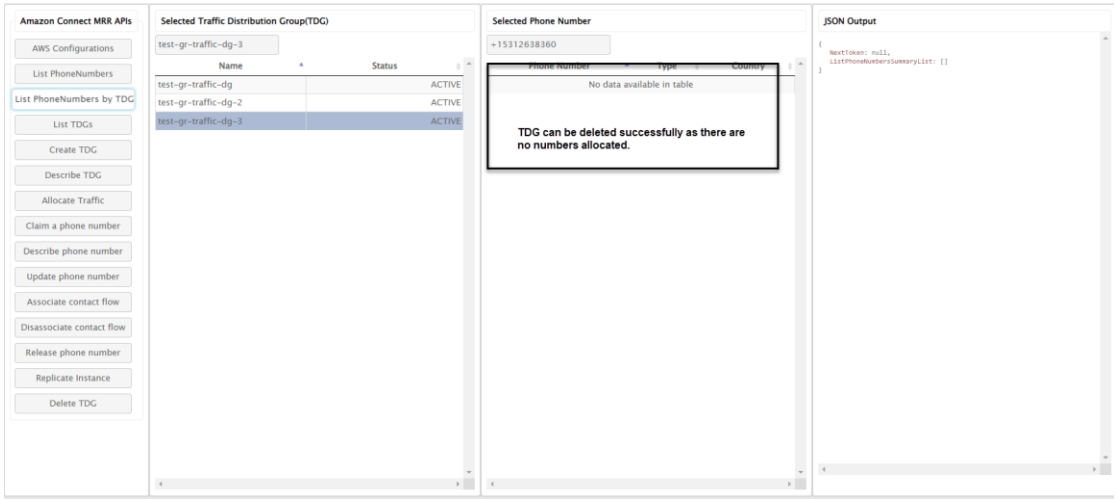


Figure 17. JSON Result of Describe TDG API

The user interface allows the creation of the TDG, whose name and optional description are entered by the user in the input field. The form can serve as a starting point for implementing failover logic in Global Resiliency for Amazon Connect. The UI hides the complexity of backend configuration and exposes it to administrators with a large range of technical skills by having a maximum of fields and a simple layout. The response the system gives in the JSON Output Pillar and TDG listing after the Create TDG action is triggered. It contains data like the name of the TDG, instance ARN, and state. Here, the displayed status is CREATION_IN_PROGRESS, indicating that group provisioning is in progress. This JSON view enables administrators to have a programmatic view of the TDG creation and ensure they are programmatically linked to the desired Amazon Connect instance. The result is also applicable in auditing and the automatic processing of work where TDG attributes are needed.

6.2.4.2. Describing TDGs

After a Traffic Distribution Group (TDG) is in the ACTIVE state, an administrator can access its entire configuration through the Describe TDG command in the web-based admin UI. The operation provides a clear view of the TDG's attributes, including the TDG name, Amazon Resource Name (ARN), instances of Amazon Connect related to it, and its status. The steps, as illustrated in Figure 19, involve selecting a TDG in the interface TDG Pillar and invoking the 'describe' action, after which a structured JSON response is returned in the JSON Output Pillar.

This feature is crucial for both pre-deployment validation and ongoing management. It helps confirm that the TDG is in the correct state before mapping phone numbers or allocating traffic percentages, which reduces the likelihood of errors in routing. The output is also stored in the backend metadata, which can be used in automation scripts, compliance auditing, and troubleshooting processes. Administrators can protect against misrouted traffic or failed failovers in multi-region scenarios by verifying the TDG state and identifiers before attempting any critical configurations. The result was a complete success. Describe TDG action. Here, the TDG, test-gr-traffic-dg, is already in the ACTIVE state and is attached to a specific Amazon Connect instance, test-dg, with its InstanceARN. Such production acts as a structural and health checkpoint, as well as a validation point, to ensure that the TDG is available to engage in routing logic and resource mapping.

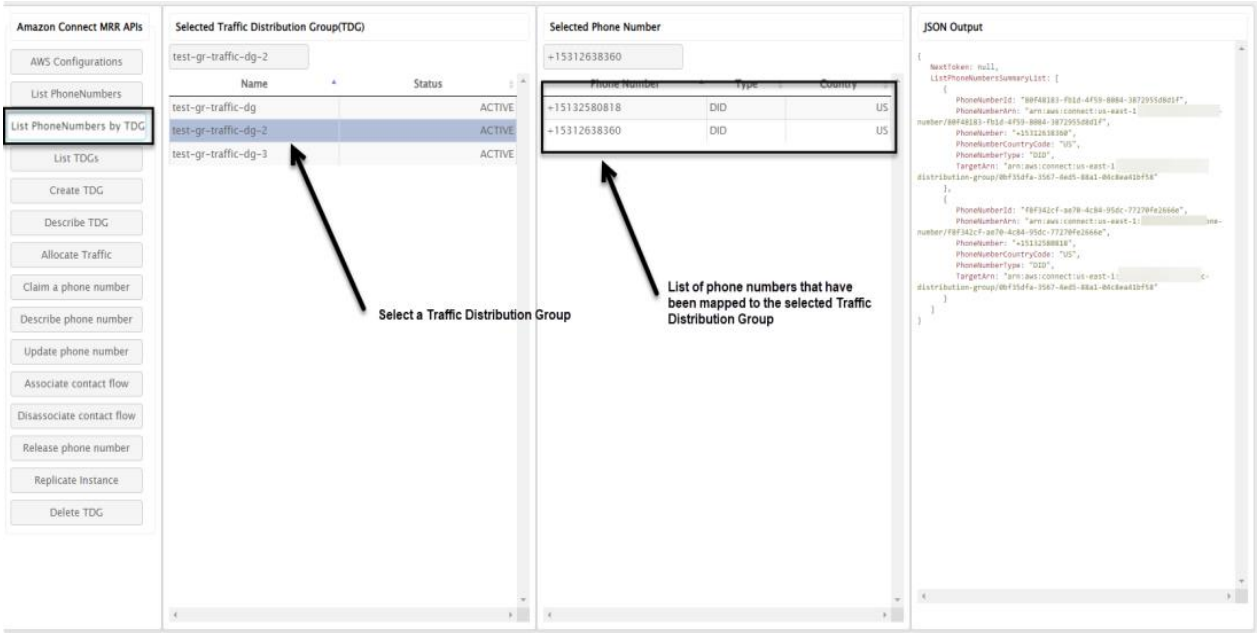


Figure 18. JSON Result of Describe TDG API

Once a phone number has been claimed and TDGs have been configured, the administrator may need to reassign the number to another Connect instance or a different TDG. Here, the action of editing the phone number, i.e., the Update Phone Number, would be important. The administrator then chooses a value under the Phone Number Pillar and then utilizes the update process. There is an option to allow the user to select the desired detail, which determines whether the number should be assigned directly to a Connect instance or indirectly through a TDG.

6.2.4.3. Updating Phone Number Mappings

Once a phone number has been claimed and TDGs have been configured, the administrator might need to assign the number to a different Connect instance or TDG. This is where the “Update Phone Number” action comes in handy. After selecting a number in the Pillar of Phone Number, the administrator then starts the process of updating. A dialogue is provided, and the user can select whether the number needs to be directly mapped to a Connect instance or whether it needs to be mapped indirectly through a TDG.

This is a dual-assigning flexibility that promotes advanced routing policies, allowing administrators to have fine control over how calls are processed: by region, availability, or even business rules. On the clicking of either of the two buttons, the update API request is posted, and the result of a successful reassignment is displayed in the JSON Output Pillar. The feedback loop of this real-time nature aids in the proper implementation of the changes and acts as a point of reference to revert to at some later date when auditing or otherwise considering a rollback is necessary.

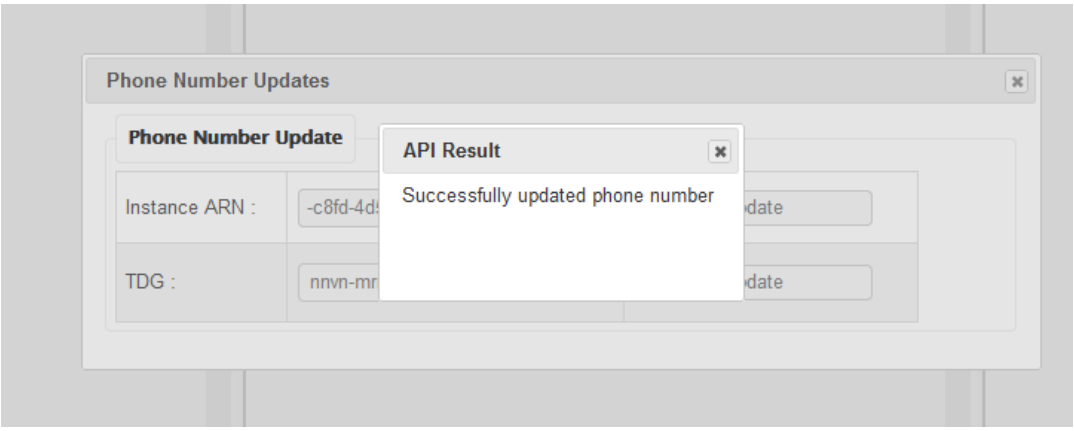


Figure 19. API Response – Successfully Updated Phone Number

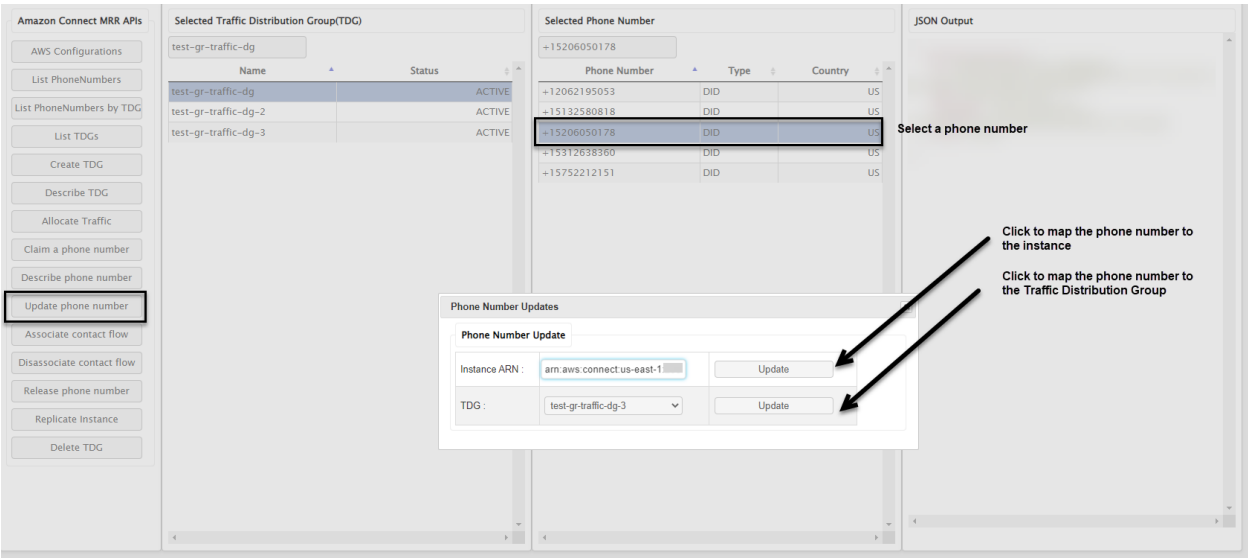


Figure 20: Update Phone Number Dialog with Mapping Options

The UI element employed to reallocate a phone number to a particular instance, either one or all of a Traffic Distribution Group. The interface is divided into two actions: updating the Instance ARN and selecting the TDG from the drop-down menu.

6.2.4.4. Reassigning and Confirming Phone Number Mappings

Once a phone number has been chosen and connected to the desired instance or Traffic Distribution Group (TDG) by clicking on the Update phone number operation, the system will generate an instant confirmation message in a dialog. The message appearing on the UI is a clear indication of the success of an update to the phone number, with the message stating that it has been successfully updated. This is a small but important feedback loop that ensures the backend API receives and performs the request as intended. It completes the circle of configuration and execution process, minimizing the possibility of uncertainty in the process of operational updates and keeps the administrators up to date regarding the success or failure of the action in real time. The API Result window appears after the successful reassignment of a phone number. A modal pop-up displays to the administrator that the number has been duly updated, either associated with a Connect instance or redirected to a TDG. This visual verification reduces the ambiguity of operations, giving the administrator confidence that the resource is available to handle the traffic routing.

6.2.4.5. Listing Phone Numbers by TDG

Amazon Connect Global Resiliency also has a process that allows you to view all numbers allocated to a specific Traffic Distribution Group (TDG). This is when contact centre administrators need to distribute the workload across regions. With the help

of the UI, where users can use the action “List PhoneNumbers by TDG”, they can retrieve all numbers that have an association with any available TDG. The system then fills in the Phone Number Pillar with the findings and presents an organized API result in the JSON Output Pillar. This feature enables the proper configuration of the traffic distribution setup and facilitates the validation of settings before implementing rebalancing or failover activities.

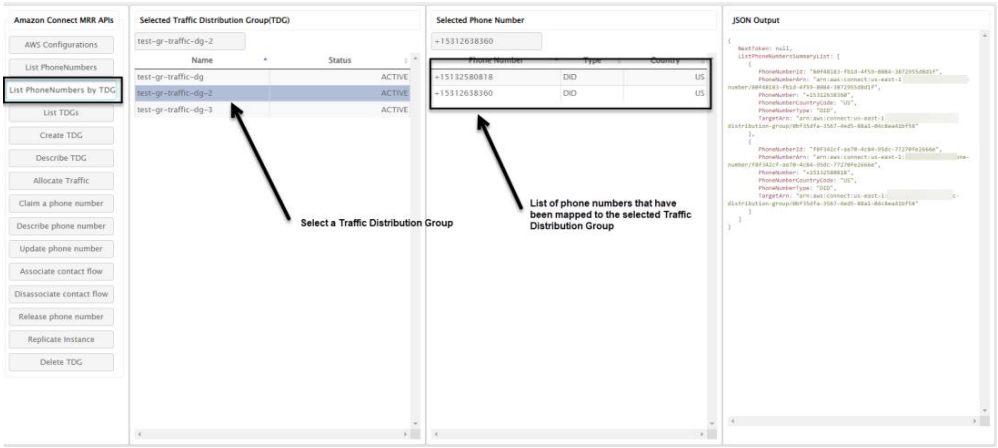


Figure 21. TDG-Linked Phone Number Listing

The result of listing the phone numbers related to the TDG test-gr-traffic-dg-2. The native interface on the left-hand displays the chosen TDG, and the Phone Number Pillar in the middle fills up with numbers that are already connected to this group. The JSON Output Pillar provides detailed metadata, including the PhoneNumberId, TargetArn, and PhoneNumberType. The view is useful in helping administrators cross-check assignments, verify the routing configuration, and track capacity in each TDG.

6.2.4.6. Allocating Traffic across Regions

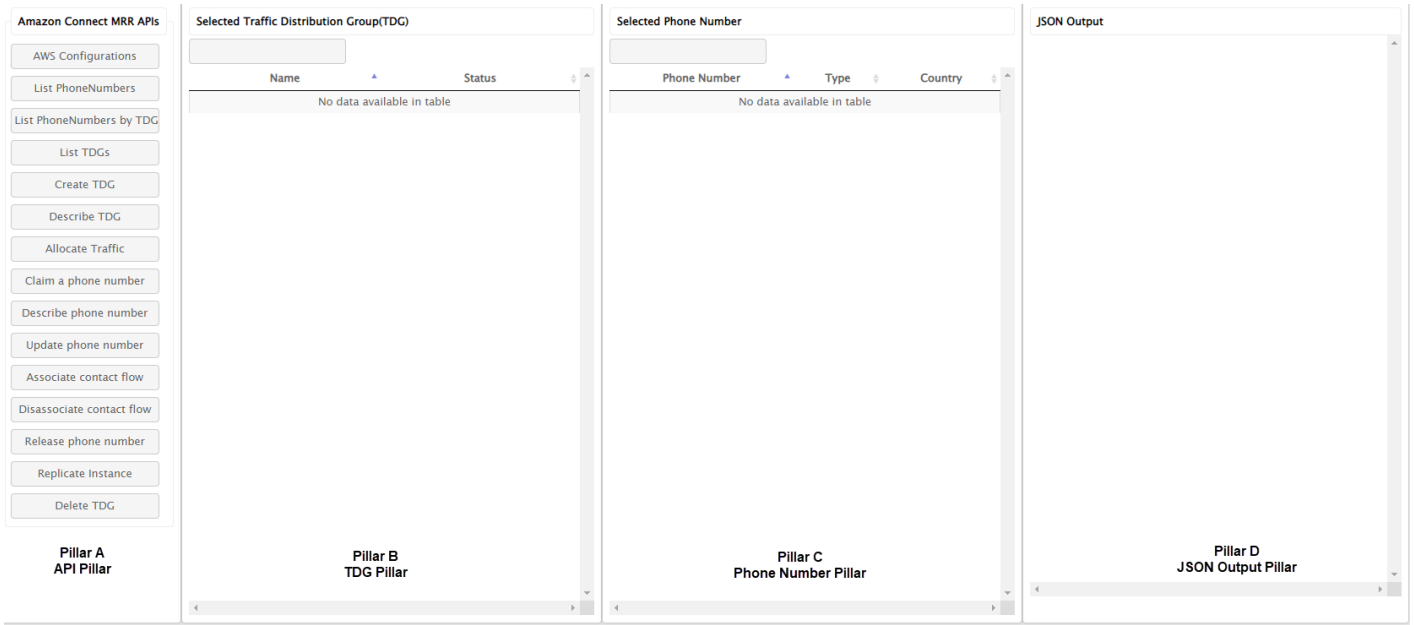


Figure 22. Allocate Traffic Percentages UI

After developing TDGs and linking them with phone numbers, the administrators can assign inbound call traffic to multiple AWS Regions through the use of the Allocate Traffic feature. This is a mandatory capability to support active-active or active-passive design across regions. The interface also presents the decision maker with the option of defining the amount of the traffic

that goes to each region in a 10 percent increment, and it gives the decision maker the ability to have fine control of how the calls are balanced.

The user selects a TDG through the TDG Pillar, clicks on the Featured page, and allocates the percentages split between primary and secondary areas, e.g., 90% to us-west-2 and 10% to us-east-1. After a submission has been made, the system calls the required API, and the JSON Output Pillar displays the new configuration. The visualization will not only verify proper execution but also provide a real-time audit trail, which can be used as a transparent approach to handling resiliency plans for failover and latency optimization. This distribution of traffic does not require the provision of infrastructure, allowing it to occur dynamically, which makes this feature invaluable during operational testing and disaster recovery scenarios.

The interface to configure traffic distribution rules to distribute traffic among the AWS Regions through the use of TDGs. The dialog traps user-set percentages and confirms the percentage after which the logic behind the routing is overridden. One-click traffic redistribution is possible because the complexity of the underlying call routing infrastructure is hidden behind a simple UI, allowing administrators to redistribute traffic loads within a few clicks. The correct JSON output serves as confirmation.

6.2.4.7. Deleting TDGs and Handling Dependencies

When there is no further need to use a Traffic Distribution Group (TDG), e.g. because of restructuring or consolidation, the API call: Delete TDG can be called by administrators. But deletion can only be made in cases where the TDG is not linked to any active phone numbers. It becomes impossible to delete a TDG that has active associations, and thereby throws a failure message and warns the user to release the numbers first.

The console through which traffic distribution rules are configured to cover multiple AWS Regions by employing TDGs. The dialog stores the user-entered percentages, and it refreshes the underlying routing logic after it is confirmed. The user interface is very simple, so that behind the scenes, the complex work of the utility in the call routing infrastructure can be hidden, thus allowing administrators to re-route traffic load in just a few clicks. The JSON output on the right-hand side confirms the checks.

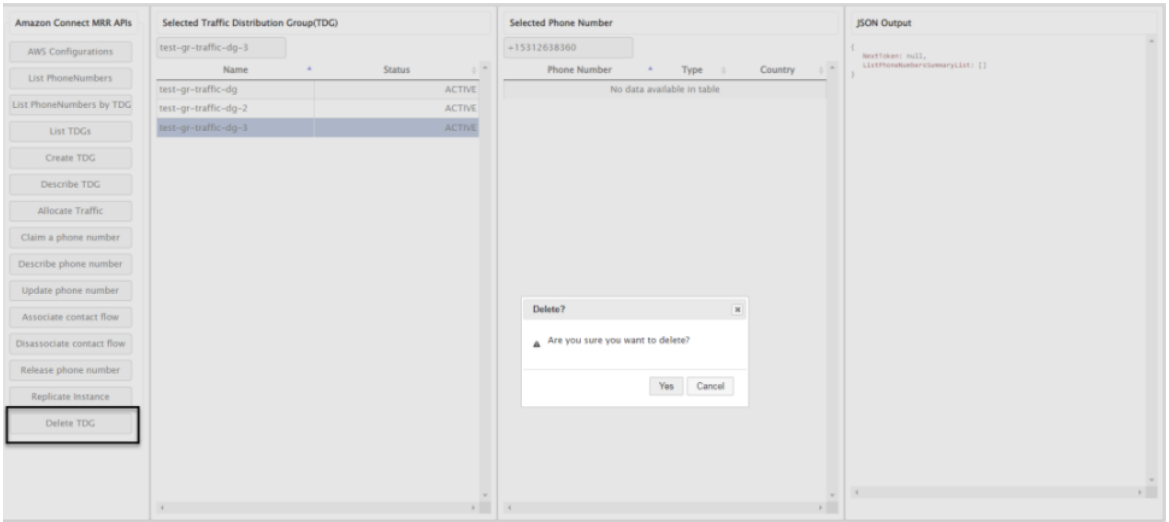


Figure 23. Allocate Traffic Percentages UI

6.2.4.8. Deleting TDGs and Handling Dependencies

When a Traffic Distribution Group (TDG) is no longer needed, for example, as a result of reorganisation or merger, an administrator may use the API labelled "Delete TDG." Deletion may, however, only be done where the TDG is not linked to any active phone numbers. When attempting to delete a TDG that is in active associations, an error message is displayed, advising the user to release the numbers before deleting the TDG.

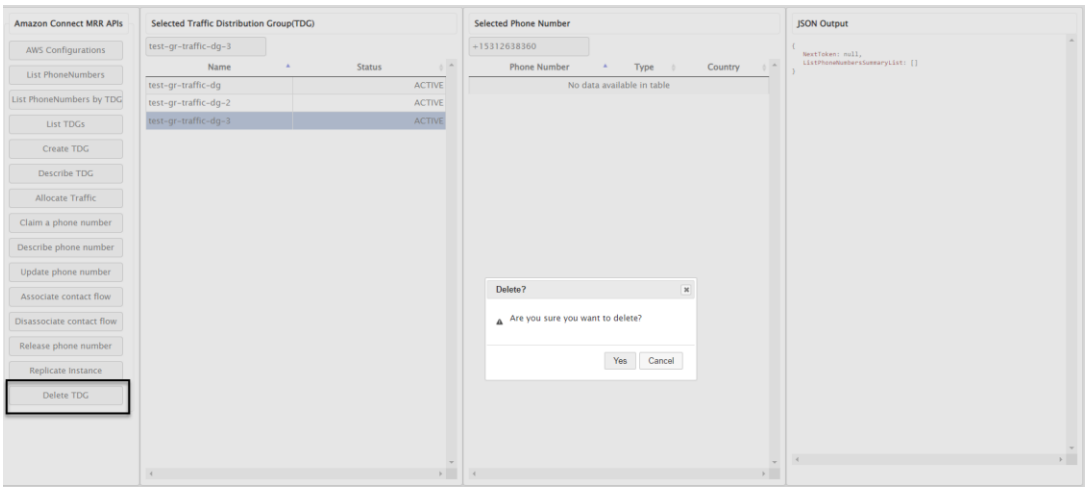


Figure 24. Delete TDG Dialog and Error Message

This two-step protection helps maintain the stability of call routing and prevents the accidental orphaning of any telephony resource. After the phone numbers have been successfully dropped using the Release phone number button, the system permits the deletion to occur. This process is displayed in the UI as a change in status (e.g., PENDING_DELETION) in the TDG Pillar, and the API returns the final response in the JSON Output Pillar. These processes of controlled deletion impose cloud-native hygiene and guard against configuration drift in the enterprise.

The screen in the TDG deletes operation. In the first case, an error can be returned if case numbers are not assigned. When the dependencies are resolved by unbinding the numbers, the TDG enters the PENDING_DELETION state, as shown in the JSON response. This sequence explains integrity checks inherent in the platform that ensure consistency in the state of global resources.

7. Amazon Connect: Global Resiliency – Onboarding Patterns

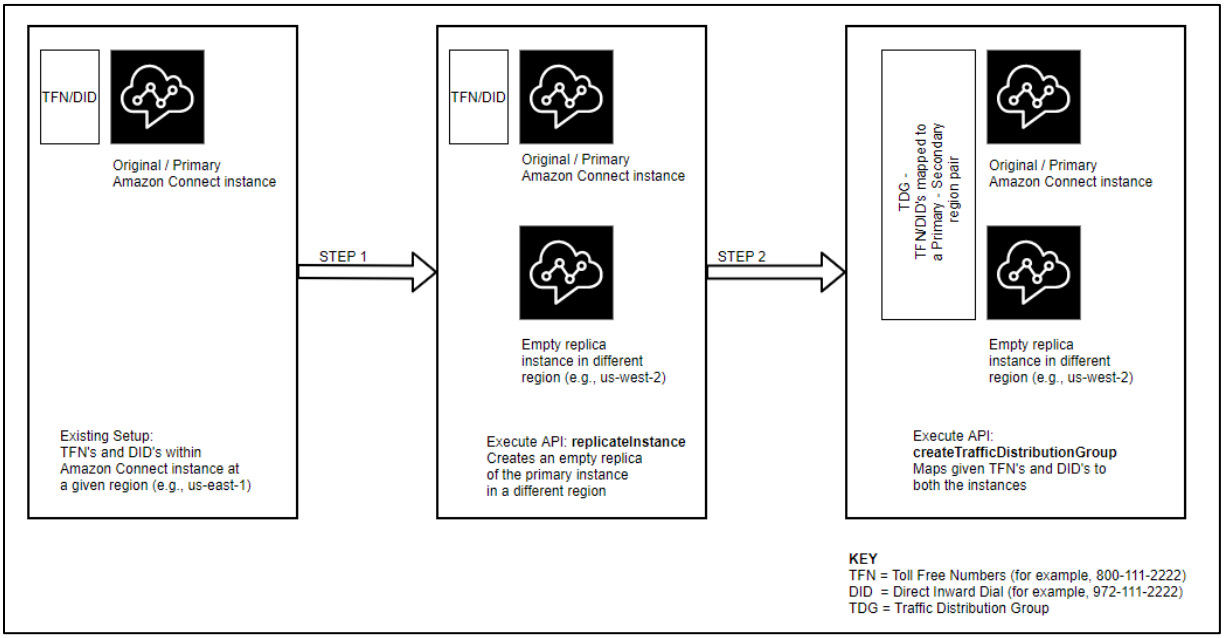


Figure 25. Amazon Connect Global Resiliency Onboarding Pattern Diagram

Amazon Connect Global Resiliency onboarding is a gated and organized workflow that must be previously coordinated with an AWS account rep to determine eligibility and applicability. Because of the capability of the feature to control the cross-region traffic routing of the mission-critical contact centers of the customers, Amazon only allows access to the customers with special pre-requirements established in regard to the preparedness of infrastructure, telephony settings, and identity management. The onboarding process generally involves the following steps: enabling service, registering instances of Amazon Connect, gaining access to API-allow-listed resources, and deploying the administrative UI using AWS CloudFormation. It also informs the organizations on how to validate their IAM policies, how to prepare SAML integrations in case they exist, and how to claim phone numbers in regulatory-compliant areas.

When organizations use this onboarding sequence built by AWS, the tool reduces the likelihood of configuration mistakes, speeds up the delivery, and provides a strong basis in terms of multi-region resiliency. The standard deployment process used to activate the Amazon Connect Global resiliency. The diagram visualizes the major steps: starting with creating API access requests and installing infrastructure in two AWS Regions, to the final process of setting up the pairing of instances and distributing traffic. This character is a reference to ensure being prepared at every point of the resiliency lifecycle and also encourages uniformity across deployments in a large-scale enterprise setup.

7.1. SAML Setup for Amazon Connect

When deploying Amazon Connect as part of a global deployment, it is important to configure a SAML-based Single Sign-On (SSO) user authentication so that the same and secure user identification endpoint is used on both the primary and replica ends. By integrating SAML, the agents and administrators can then use their enterprise credentials through a central Identity Provider (IdP), which facilitates uniform access control and simplified user management. Amazon Connect instances should be configured with the same SAML parameters to ensure seamless functioning during failovers, including the selection of trust relationships, role mappings, and directories. This will guarantee that other agents who are redirected to the secondary region can authorize themselves without waiting or losing their access privileges. Failed log-ins or mismatched roles are possible, with potential real-time client-service effects in the case of misaligned configurations. This means that the resiliency validation should include a comprehensive testing of SAML in both regions.

7.2. Resource Design: Amazon CloudFront and Amazon S3

Amazon CloudFront and S3 appear as a combined solution to build the backbone of the admin UI layer of Amazon Connect Global Resiliency. CloudFormation deploys a custom static web interface using the assets we are going to store (HTML, JavaScript, CSS, etc) into a specific S3 bucket and then distributes them worldwide using CloudFront. This configuration provides scalability, low latency, and a global scope for administrative processes, including instance replication and traffic management. However, because CloudFront distributions are default, publicly accessible, hardening of security is essential. Controls such as Origin Access Control (OAC), HTTPS, and access control using IAM or signed URLs should be implemented by administrators. Additionally, it is essential to activate CloudFront logs and verify that the CDN health and S3 connection are functioning properly, ensuring the interface operates securely and with high availability in production. These layer misconfigurations may also deny access to the UI or open it to unauthorized operations, compromising the resilience strategy.

8. Performance Evaluation and Observations

Critical assessments of Global Resiliency solutions in Amazon Connect can highlight the role of Amazon Connect as an effective way to run business applications and solutions in a relational dependency, supporting the business functions and operations of a company at an enterprise level. The performance measurements that have been recorded in the implementation simulation and in testing phases show that the system is designed to tune high availability, fast provisioning, and failover between regions without impeding other features. All parts of the architecture, including the setup of instances, real-time traffic migration, etc., have been examined to quantify resilience and responsiveness across different workloads and configurations.

8.1. Instance Creation Timelines

Replicating a primary Amazon Connect instance to use as a secondary instance in a different region is expected to take 5-8 minutes, depending on the workload and service quotas in the desired region. The operation of creating an instance is triggered over the operation of replicating an instance using the API, and these operations are asynchronous. The AWS Management Console currently records the instance's status in the status field as 'Pending'. Once provisioning is complete, the instance has all the requirements to be fully operational and can now be attached to a Traffic Distribution Group (TDG). This timeline shows how efficient the service is in its ability to deploy large-scale and geographically distributed environments, and this is essential when it comes to time-sensitive deployments.

8.2. API Responsiveness and UI Effectiveness

The user interface, created with AWS CloudFormation and served by Amazon CloudFront, responded well to interactions due to its web-based nature. Phone number claiming, TDG creation, and instance updates, as well as all other actions in the API, took a couple of seconds to complete, with a real-time JSON response displayed in the UI on a separate output panel. The explicitly divided interface with Pillars A to D (API Actions, TDG List, Phone Number Management, and JSON Output) also makes it more usable and operationally transparent. Users are allowed to run a series of operations without the need to reload or reauthenticate, confirming the efficiency of the UI when used by administrators who deal with critical configuration settings.

8.3. Latency and Traffic Rebalancing Efficiency

Among the most influential experimental features, one can distinguish the traffic reallocation mechanism with TDGs. Modification of the traffic portion in two areas (ex, 70 percent in us-west-2 and 30 percent in us-east-1) was pretty much instant, and the adjustment specialized in the UI and the backend API. Since this functionality is based on managed Amazon Connect infrastructure, the latency between applying changes to a TDG and seeing the new traffic behavior was minimal. Such ability is critical when it comes to the operations teams that need to realign the workload while there is a partial outage or when workloads need to be optimized on a per-geographical route, based on the volume of calls made.

8.4. Resilience during Failover Scenarios

Continuous performance of the system in simulated failover was also a key highlight of the system during the test. Calls were effectively diverted to the secondary region with no manual intervention into the system other than initial reconfiguration of the TDG when the primary region was disabled or traffic allocation to it was set to 0 percent. This proved the feasibility of using the Global Resiliency of Amazon Connect to achieve business continuity and disaster recovery. The failover process, in the event of a failure of any one of the agents, resulted in no disruption of user authentication or agent workflow when combined with proper IAM policies (additional authentication) and SAML configuration. This aspect demonstrates the power of the architecture even in adverse situations

.

9. Security and Compliance Considerations

9.1. Ensuring Security and Compliance in Global Contact Center Deployments

The security-first requirement in deploying Amazon Connect, with global resiliency across different regions, is a paramount concern when handling sensitive customer data and dynamic failover processes. Hosting of administrative interfaces in Amazon CloudFront presents it as a possible method of attack when not securely configured. Origin Access Control (OAC) should be used to regulate access to CloudFront distributions by users outside of organizations, and HTTPS should be enforced; signed URLs should be used or simple AWS WAF rules. The static assets ought not to be subject to S3 direct access, and access must be logged and monitored. Moreover, the safe treatment of credentials using such services as AWS Secrets Manager or Amazon Cognito will guarantee that any potentially vulnerable API keys are never hardcoded or openly exposed.

9.2. Strengthening IAM and Privacy Controls across Regions

One of the foundations of this type of security is careful management of IAM roles. Resiliency components that have been deployed using templates that have pre-configured roles created by CloudFormation should be assessed and determined to meet least privilege requirements. Access to the minimum required actions should be strictly restricted, and sensitive operations like

redirection of traffic, replication of instances, etc., should be accessible only via well-audited, role-based access control. Access control can also be reinforced by integrating it with AWS IAM Identity Centre to require SSO mechanisms for user authentication. In the meantime, privacy regulations such as GDPR and HIPAA must be observed in a cross-region configuration. Organizations need to make sure that telephony traffic or PII is not flowing unintentionally across regional boundaries, which requires encryption and regional storage as well as newer privacy disclosures to comply with data sovereignty requirements.

10. Cost and Resource Management

10.1. Managing Costs in a Multi-Region Amazon Connect Deployment

Although Amazon Connect Global Resiliency provides benefits in increased service and disaster recovery, it also creates new operational costs, as the infrastructure is doubled in regions. The CloudFront distributions, Amazon Connect instances, IAM roles and S3 buckets are resources, and their costs, dependent on the level of usage, are not relevant. Accounts that are unused or on standby for phone numbers can also incur monthly fees. Organizations can monitor such resources with AWS Cost Explorer, Budgets, and tagging policies that help associate costs with an environment, project, or team. To ensure the resilience and affordability of data-driven decisions, it is crucial to gain a clear understanding of resource behaviour.

10.2. Optimization and Cleanup Strategies for Sustainable Operations

The most successful cost optimization procedure is complemented with automation and active governance. When it comes to Amazon S3, there is a need to apply lifecycle policies to auto-delete or store obsolete web assets. Minimizing the cost of using CloudFront can be achieved through restricting geographic spread, facilitating caching, and avoiding unnecessary logging without any compliance needs. The Amazon Connect configuration should be set up to consume only the necessary phone numbers, and the actual traffic change steps in DR testing should be restricted to prevent high-volume invoicing in the backup areas. The other important issue is cleanup; administrators must conform to a good teardown of the resources after testing via CloudFormation stack deletion and follow-ups manually to make the phone numbers and IAM roles available. Such operations not only help lower operational costs but also contribute to creating a safe and well-maintained AWS environment.

11. Limitations and Challenges

11.1. Recognizing the Challenges of Global Resiliency Deployments

Although it can be used to improve high availability and disaster recovery, Amazon Connect Global Resiliency also has a few drawbacks that may significantly impede its future implementation among globally distributed enterprises. The main constraint is geographic locational variation, which is currently limited to the US-East-1 and US-West-2 regions. This limits the deployment to the organizations working in Europe, Asia-Pacific or other jurisdictions where data residency and latency are strict. Companies that operate cross-regionally must either wait longer for wider regional support to be offered or create bespoke workarounds, which can be costly and time-consuming.

11.2. Operational and Security Constraints

Along with local constraints, the need to manually interfere with critical processes to release phone numbers, provide access keys, configure SAML, etc., affects automation and reactiveness. These manual procedures introduce latency and increase the possibility of human error, particularly in urgent failover situations. Moreover, access to Global Resiliency services is restricted by the AWS allow-listing, i.e., it should be approved in advance, which can impede the onboarding process. This presents a major challenge to the implementation of secure, least-privilege access coupled with complicated IAM implementations. Finally, the administrative UI might be accessed by other people without requiring legitimate credentials in case it is not adequately secured via CloudFront security features and, as a consequence, gives malicious actors control over the underlying infrastructure. Tight security measures, such as WAF rules, OAC, IP allow-listing, and temporary credential handling, should also be implemented to reduce these risks.

12. Conclusion

Amazon Connect Global Resiliency represents a revolutionary evolution in cloud-based contact centre design, featuring automated and API-based failover across AWS Regions. Organizations are now able to architect highly available contact center

solutions that can survive regional disruptions without having to degrade the service provided to customers due to the use of Traffic Distribution Groups (TDGs), the ability to replicate instances, and a modular web-based administration UI. The incorporation of CloudFormation, SAML-based authentication, IAM best practices, and phone number management facilitating mechanisms makes the resiliency model not only technically sufficient but also feasible in terms of operation and deployment practice in the real world for enterprise usage.

Despite the fact that the current implementation is confined to several regions within the U.S. and needs some manually-based configurations, the framework has created the platform upon which a resiliency strategy can be scalable and extensible. Amazon Connect Global Resiliency is capable of helping minimizing the risks of downtime and serving mission-critical communication channels when deployed, configured, and managed properly to ensure the resiliency and safety of the communication channels between various missions, teams, and customers located in different geographical locations. With the current pace of AWS innovation around this offering (support for global regions, complex automation, and AI-driven routing), the solution could become a new standard in next-generation cloud-native contact centre architecture.

References

[1] Perz, S. G., Shenkin, A., Barnes, G., Cabrera, L., Carvalho, L. A., & Castillo, J. (2012). Connectivity and resilience: a multidimensional analysis of infrastructure impacts in the southwestern Amazon. Social indicators research, 106, 259-285.

[2] Ruiz Agudelo, C. A., Mazzeo, N., Díaz, I., Barral, M. P., Piñeiro, G., Gadino, I., ... & Acuña Posada, R. J. (2020). Land use planning in the Amazon basin: challenges from resilience thinking.

[3] AWS Contact Center Blog, "Build a multi-region resilient contact center with Amazon Connect Global Resiliency," Amazon Web Services, Nov. 29, 2022. [Online]. Available: https://aws.amazon.com/blogs/contact-center/build-a-multi-region-resilient-contact-center-with-amazon-connect-global-resiliency/

[4] Ahlström, A., Canadell, J. G., Schurgers, G., Wu, M., Berry, J. A., Guan, K., & Jackson, R. B. (2017). Hydrologic resilience and Amazon productivity. Nature communications, 8(1), 387.

[5] Rao, P., Lin, D., Bertino, E., Li, N., & Lobo, J. (2011). Fine-grained integration of access control policies. Computers & Security, 30(2-3), 91-107.

[6] Tovmasyan, K. (2020). Mastering AWS CloudFormation: Plan, develop, and deploy your cloud infrastructure effectively using AWS CloudFormation. Packt Publishing Ltd.

[7] Ristov, S., Kimovski, D., & Fahringer, T. (2022). Fascinating resilience for serverless function choreographies in federated clouds. IEEE Transactions on Network and Service Management, 19(3), 2440-2452.

[8] Kulkarni, S. G., Liu, G., Ramakrishnan, K. K., & Wood, T. (2019, July). Living on the edge: Serverless computing and the cost of failure resiliency. In 2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) (pp. 1-6). IEEE.

[9] Nehme, A., Jesus, V., Mahbub, K., & Abdallah, A. (2018, November). Fine-grained access control for microservices. In International Symposium on Foundations and Practice of Security (pp. 285-300). Cham: Springer International Publishing.

[10] Trifan, L. (2013). Resiliency in Distributed Workflow Systems for Numerical Applications (Doctoral dissertation, Université de Grenoble).

[11] Sarkar, A., & Shah, A. (2018). Learning AWS: Design, build, and deploy responsive applications using AWS Cloud components. Packt Publishing Ltd.

[12] Ho, J. W., Liu, D., Wright, M., & Das, S. K. (2009). Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks. Ad Hoc Networks, 7(8), 1476-1488.

[13] Varia, J. (2011). Best practices in architecting cloud applications in the AWS cloud. Cloud Computing: Principles and Paradigms, 457-490.

[14] Wilkins, M. (2019). Learning Amazon Web Services (AWS): A hands-on guide to the fundamentals of AWS Cloud. Addison-Wesley Professional.

[15] Baron, J., Baz, H., Bixler, T., Gaut, B., Kelly, K. E., Senior, S., & Stamper, J. (2016). AWS certified solutions architect official study guide: associate exam. John Wiley & Sons.

[16] Acharya, K. (2022). Assessing the Resilience of Adaptive Intrusion Prevention Systems in SaaS-Driven E-Retail Ecosystems. Journal of Emerging Cloud Technologies and Cross-Platform Integration Paradigms, 6(12), 1-11.

[17] Bleikertz, S., Schunter, M., Probst, C. W., Pendarakis, D., & Eriksson, K. (2010, October). Security audits of multi-tier virtual infrastructures in public infrastructure clouds. In Proceedings of the 2010 ACM workshop on Cloud computing security workshop (pp. 93-102).

[18] Watanabe, C., Akhtar, W., Tou, Y., & Neittaanmäki, P. (2022). A new perspective of innovation toward a non-contact society-Amazon's initiative in pioneering growing seamless switching. Technology in Society, 69, 101953.

[19] Raheja, Y., Borgese, G., & Felsen, N. (2018). Effective DevOps with AWS: Implement continuous delivery and integration in the AWS environment. Packt Publishing Ltd.

[20] Wadia, Y. (2018). AWS Administration-the Definitive Guide: Design, Build, and Manage Your Infrastructure on Amazon Web Services. Packt Publishing Ltd.