User Guide

AWS Control Tower

AWS Control Tower: User Guide

Table of Contents

What Is AWS Control Tower?

AWS Control Tower offers a straightforward way to set up and govern an AWS multi-account environment, following prescriptive best practices. AWS Control Tower *orchestrates* the capabilities of several other AWS services, including AWS Organizations, AWS Service Catalog, and AWS IAM Identity Center, to build a landing zone in less than an hour. Resources are set up and managed on your behalf.

AWS Control Tower orchestration extends the capabilities of AWS Organizations. To help keep your organizations and accounts from *drift*, which is divergence from best practices, AWS Control Tower applies controls (sometimes called *guardrails*). For example, you can use controls to help ensure that security logs and necessary cross-account access permissions are created, and not altered.

If you are hosting more than a handful of accounts, it's beneficial to have an orchestration layer that facilitates account deployment and account governance. You can adopt AWS Control Tower as your primary way to provision accounts and infrastructure. With AWS Control Tower, you can more easily adhere to corporate standards, meet regulatory requirements, and follow best practices.

AWS Control Tower enables end users on your distributed teams to provision new AWS accounts quickly, by means of configurable account templates in Account Factory. Meanwhile, your central cloud administrators can monitor that all accounts are aligned with established, company-wide compliance policies.

In short, AWS Control Tower offers the easiest way to set up and govern a secure, compliant, multi-account AWS environment based on best practices established by working with thousands of enterprises. For more information about working with AWS Control Tower and the best practices outlined in the AWS multi-account strategy, see AWS multi-account strategy: Best practices guidance.

Features

AWS Control Tower has the following features:

- **Landing zone** – A landing zone is a well-architected, multi-account environment that's based on security and compliance best practices. It is the enterprise-wide container that holds all of your organizational units (OUs), accounts, users, and other resources that you want to be subject to compliance regulation. A landing zone can scale to fit the needs of an enterprise of any size.

- **Controls –** A control (sometimes called a *guardrail*) is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. Three kinds of controls exist: *preventive*, *detective*, and *proactive*. Three categories of guidance apply to controls: *mandatory*, *strongly recommended*, or *elective*. For more information about controls, see How controls work.
- **Account Factory –** An Account Factory is a configurable account template that helps to standardize the provisioning of new accounts with pre-approved account configurations. AWS Control Tower offers a built-in Account Factory that helps automate the account provisioning workflow in your organization. For more information, see Provision and manage accounts with Account Factory.
- **Dashboard –** The dashboard offers continuous oversight of your landing zone to your team of central cloud administrators. Use the dashboard to see provisioned accounts across your enterprise, controls enabled for policy enforcement, controls enabled for continuous detection of policy non-conformance, and noncompliant resources organized by accounts and OUs.

How AWS Control Tower interacts with other AWS services

AWS Control Tower is built on top of trusted and reliable AWS services including AWS Service Catalog, AWS IAM Identity Center, and AWS Organizations. For more information, see Integrated services.

You can incorporate AWS Control Tower with other AWS services into a solution that helps you migrate your existing workloads to AWS. For more information, see How to take advantage of AWS Control Tower and CloudEndure to migrate workloads to AWS.

Configuration, Governance, and Extensibility

- *Automated account configuration:* AWS Control Tower automates account deployment and enrollment by means of an Account Factory (or “vending machine”), which is built as an abstraction on top of provisioned products in AWS Service Catalog. The Account Factory can create and enroll AWS accounts, and it automates the process of applying controls and policies to those accounts. For more information about creating and provisioning accounts, see Methods of provisioning.
- *Centralized governance:* By employing the capabilities of AWS Organizations, AWS Control Tower sets up a framework that ensures consistent compliance and governance across your multi-account environment. The AWS Organizations service provides essential capabilities for managing a multi-account environment, including central governance and management of

accounts, account creation from AWS Organizations APIs, service control policies (SCPs), and resource control policies (RCPs).

- *Extensibility:* You can build or extend your own AWS Control Tower environment by working directly in AWS Organizations, as well as in the AWS Control Tower console. You can see your changes reflected in AWS Control Tower after you register your existing organizations and enroll your existing accounts into AWS Control Tower. You can update your AWS Control Tower landing zone to reflect your changes. If your workloads require further advanced capabilities, you can leverage other AWS partner solutions along with AWS Control Tower.

Are You a First-Time User of AWS Control Tower?

If you’re a first-time user of this service, we recommend that you read the following:

1. If you need more information about how to plan and organize your landing zone, see Plan your AWS Control Tower landing zone and AWS multi-account strategy for your AWS Control Tower landing zone.
2. If you’re ready to create your first landing zone, see Getting started with AWS Control Tower.
3. For information on drift detection and prevention, see Detect and resolve drift in AWS Control Tower.
4. For security details, see Security in AWS Control Tower.
5. For information on updating your landing zone and member accounts, see Configuration update management in AWS Control Tower.

How AWS Control Tower works

This section describes at a high level how AWS Control Tower works. Your landing zone is a well-architected multi-account environment for all of your AWS resources. You can use this environment to enforce compliance regulations on all of your AWS accounts.

Structure of an AWS Control Tower Landing Zone

The structure of a landing zone in AWS Control Tower is as follows:

- **Root** – The parent that contains all other OUs in your landing zone.

- **Security OU –** This OU contains the Log Archive and Audit accounts. These accounts often are referred to as *shared accounts*. When you launch your landing zone, you can choose customized names for these shared accounts, and you have the option to bring existing AWS accounts into AWS Control Tower for security and logging. However, these cannot be renamed later, and existing accounts cannot be added for security and logging after initial launch.
- **Sandbox OU –** The Sandbox OU is created when you launch your landing zone, if you enable it. This and other registered OUs contain the enrolled accounts that your users work with to perform their AWS workloads.
- **IAM Identity Center directory –** By default, this directory houses your IAM Identity Center users. It defines the scope of permissions for each IAM Identity Center user. Optionally, you can choose to self-manage your identity and access control. For more information, see Working with AWS IAM Identity Center and AWS Control Tower.
- **IAM Identity Center users –** These are the identities that your users can assume to perform their AWS workloads in your landing zone.

What happens when you set up a landing zone

When you set up a landing zone, AWS Control Tower performs the following actions in your management account on your behalf:

- Creates two AWS Organizations organizational units (OUs): Security, and Sandbox (optional), contained within the organizational root structure.
- Creates or adds two shared accounts in the Security OU: the Log Archive account and the Audit account.
- Creates a cloud-native directory in IAM Identity Center, with preconfigured groups and single sign-on access, if you choose the default AWS Control Tower configuration, or it allows you to self-manage your identity provider.
- Applies all mandatory, preventive controls to enforce policies.
- Applies all mandatory, detective controls to detect configuration violations.
- *Preventive controls are not applied to the management account.*
- Except for the management account, controls are applied to the organization as a whole.

Safely Managing Resources Within Your AWS Control Tower Landing Zone and Accounts

- When you create your landing zone, a number of AWS resources are created. To use AWS Control Tower, you must not modify or delete these AWS Control Tower managed resources outside of the supported methods described in this guide. Deleting or modifying these resources will cause your landing zone to enter an unknown state. For details, see Guidance for creating and modifying AWS Control Tower resources
- When you enable optional controls (those with *strongly recommended or elective* guidance), AWS Control Tower creates AWS resources that it manages in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so can result in the controls entering an unknown state.

What are the shared accounts?

In AWS Control Tower, the shared accounts in your landing zone are provisioned during setup: the management account, the log archive account, and the audit account.

What is the management account?

This is the account that you created specifically for your landing zone. This account is used for billing for everything in your landing zone. It's also used for Account Factory provisioning of accounts, as well as to manage OUs and controls.

> **Note**
>
> It is not recommended to run any type of production workloads from an AWS Control Tower management account. Create a separate AWS Control Tower account to run your workloads.

For more information, see Management account.

What is the log archive account?

This account works as a repository for logs of API activities and resource configurations from all accounts in the landing zone.

For more information, see Log archive account.

What is the audit account?

The audit account is a restricted account that's designed to give your security and compliance teams read and write access to all accounts in your landing zone. From the audit account, you have programmatic access to review accounts, by means of a role that is granted to Lambda functions only. The audit account does not allow you to log in to other accounts manually. For more information about Lambda functions and roles, see Configure a Lambda function to assume a role from another AWS account.

For more information, see Audit account.

How controls work

A control is a high-level rule that provides ongoing governance for your overall AWS environment. Each control enforces a single rule, and it's expressed in plain language. You can change the elective or strongly recommended controls that are in force, at any time, from the AWS Control Tower console or the AWS Control Tower APIs. Mandatory controls are always applied, and they can't be changed.

Preventive controls prevent actions from occurring. For example, the elective control called **Disallow Changes to Bucket Policy for Amazon S3 Buckets** (Previously called **Disallow Policy Changes to Log Archive**) prevents any IAM policy changes within the log archive shared account. Any attempt to perform a prevented action is denied and logged in CloudTrail. The resource is also logged in AWS Config.

Detective controls detect specific events when they occur and log the action in CloudTrail. For example, the strongly recommended control called **Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances** detects whether an unencrypted Amazon EBS volume is attached to an EC2 instance in your landing zone.

Proactive controls check whether resources are compliant with your company policies and objectives, before the resources are provisioned in your accounts. If the resources are out of compliance, they are not provisioned. Proactive controls monitor resources that would be deployed in your accounts by means of AWS CloudFormation templates.

For those who are familiar with AWS: In AWS Control Tower preventive controls are implemented with service control policies (SCPs) and resource control policies (RCPs). Detective controls are implemented with AWS Config rules. Proactive controls are implemented with AWS CloudFormation hooks.

Related Topics

- Detect and resolve drift in AWS Control Tower

How AWS Control Tower works with StackSets

AWS Control Tower uses AWS CloudFormation StackSets to set up resources in your accounts, by default. Each stack set has StackInstances that correspond to accounts, and to AWS Regions per account. AWS Control Tower deploys one stack set instance per account and Region.

AWS Control Tower applies updates to certain accounts and AWS Regions selectively, based on AWS CloudFormation parameters. When updates are applied to some stack instances, other stack instances may be left in **Outdated** status. This behavior is expected and normal.

When a stack instance goes into **Outdated** status, it usually means that the stack corresponding to that stack instance is not aligned with the latest template in the stack set. The stack remains in the older template, so it might not include the latest resources or parameters. The stack is still completely usable.

Here's a quick summary of what behavior to expect, based on AWS CloudFormation parameters that are specified during an update:

If the stack set update includes changes to the template (that is, if the `TemplateBody` or `TemplateURL` properties are specified), or if the `Parameters` property is specified, AWS CloudFormation marks all stack instances with a status of **Outdated** prior to updating the stack instances in the specified accounts and AWS Regions. If the stack set update does not include changes to the template or parameters, AWS CloudFormation updates the stack instances in the specified accounts and Regions, while leaving all other stack instances with their existing stack instance status. To update all of the stack instances associated with a stack set, do not specify the `Accounts` or `Regions` properties.

For more information, see Update Your Stack Set in the AWS CloudFormation User Guide.

Terminology

Here's a quick review of some terms you'll see in the AWS Control Tower documentation.

First, it's good to know that AWS Control Tower shares a lot of terminology with the AWS Organizations service, including the terms *organization* and *organizational unit (OU)*, which appear throughout this document.

- For more information about organizations and OUs, see AWS Organizations terminology and concepts. If you're new to AWS Control Tower, that terminology is a good place to begin.
- AWS Organizations is an AWS service that helps you centrally govern your environment as you grow and scale your workloads on AWS. AWS Control Tower relies on AWS Organizations to create accounts, to enforce preventive controls at the OU level, and to provide centralized billing.
- An AWS Account Factory account is an AWS account provisioned using Account Factory in AWS Control Tower. Sometimes, Account Factory is referred to informally as a "vending machine" for accounts.
- Your AWS Control Tower home Region is the AWS Region in which your AWS Control Tower landing zone was deployed. You can view your home Region in your landing zone settings.
- AWS Service Catalog allows you to manage commonly deployed IT services, centrally. In the context of this document, Account Factory uses AWS Service Catalog to provision new AWS accounts, including accounts from customized blueprints.
- AWS CloudFormation StackSets are a type of resource that extends the functionality of stacks so that you can create, update, or delete stacks across multiple accounts and Regions with a single operation and a single CloudFormation template.
- A stack instance is a reference to a stack in a target account within a Region.
- A stack is a collection of AWS resources that you can manage as a single unit.
- An aggregator is an AWS Config resource type that collects AWS Config configuration and compliance data from multiple accounts and Regions within the organization, allowing you to view and query this compliance data within a single account.
- A conformance pack is a collection of AWS Config rules and remediation actions that can be deployed as a single entity in an account and a Region, or across an organization in AWS Organizations. You can use a conformance pack to help customize your AWS Control Tower environment. For technical blogs that provide more details, see Related information.
- A baseline in AWS Control Tower is a group of resources and specific configurations that you can apply to a target. The most common baseline target may be an organizational unit (OU). For

example, the baseline called `AWSControlTowerBaseline` is available to help register your OUs with AWS Control Tower. During landing zone setup and update, the baseline target may be a shared account, or a specific setting for the landing zone as a whole.

- **Blueprint:** A blueprint is an artifact that encapsulates some metadata, which describes infrastructure components that are deployed within an account. For example, an AWS CloudFormation template can serve as a blueprint for an AWS Control Tower account.
- **Drift:** A change in a resource installed by and configured by AWS Control Tower. Resources without drift enable AWS Control Tower to function properly.
- **Non-compliant resource:** A resource that is in violation of an AWS Config rule that defines a particular detective control.
- **Shared account:** One of the three accounts that AWS Control Tower creates automatically when you set up your landing zone: the management account, the log archive account, and the audit account. You can choose customized names for the log archive account and the audit account, during setup.
- **Member account:** A member account belongs to the AWS Control Tower organization. The member account can be *enrolled* or *unenrolled* in AWS Control Tower. When a registered OU contains a mix of enrolled and unenrolled accounts:
  - Preventive controls enabled on the OU apply to all accounts within it, including unenrolled ones. This is true because preventive controls are enforced with SCPs at the OU level, not the account level. For more information, see Inheritance for service control policies in the AWS Organizations documentation.
  - Detective controls enabled on the OU do not apply to unenrolled accounts.

  An account can be a member of only one organization at a time, and its charges are billed to the management account for that organization. A member account can be moved to the root container of an organization.
- **AWS account:** An AWS account acts as a resource container and resource isolation boundary. An AWS account can be associated with billing and payment. An AWS account is different than a user account (sometimes called an IAM user account) in AWS Control Tower. Accounts created through the Account Factory provisioning process are AWS accounts. AWS accounts also can be added to AWS Control Tower by means of the account enrollment or OU registration process.
- **Control:** A control (also known as a *guardrail*) is a high-level rule that provides ongoing governance for your overall AWS Control Tower environment. Each control enforces a single rule. Preventive controls are implemented with SCPs. Detective controls are implemented with AWS

Config rules. Proactive controls are implemented with AWS CloudFormation hooks. For more information, see How controls work.

- **Control Catalog:** The AWS Control Tower control catalog is the compendium of all controls that are available through AWS Control Tower, in the console and APIs. It formerly was called the Control Library. We aligned the terminology to the name of the namespace, *controlcatalog*.
- **Landing zone:** A landing zone is a cloud environment that offers a recommended starting point, including default accounts, account structure, network and security layouts, and so forth. From a landing zone, you can deploy workloads that utilize your solutions and applications.
- **Framework:** A framework is a specific regulatory standard or industry requirement. In the Control Catalog ontology, a framework is represented by a Standard control. For more information, see the Control Catalog Ontology overview.
- **Nested OU:** A nested OU in AWS Control Tower is an OU contained within another OU. A nested OU can have exactly one parent OU, and each account can be a member of exactly one OU. Nested OUs create a hierarchy. When you attach a policy to one of the OUs in the hierarchy, it flows down and affects all the OUs and accounts beneath it. A nested OU hierarchy in AWS Control Tower can be a maximum of five levels deep.
- **Parent OU:** The OU immediately above the current OU in the hierarchy. Each OU can have exactly one parent OU.
- **Child OU:** Any OU below the current OU in the hierarchy. An OU can have many child OUs.
- **OU hierarchy:** In AWS Control Tower, the hierarchy of nested OUs can have up to five levels. The order of nesting is referred to as **Levels**. The top of the hierarchy is designated as **Level 1**.
- **Top-level OU:** A top-level OU is any OU that's directly under the Root, not the Root itself. The Root is not considered an OU.
- **Governed:** A governed Region is managed and controlled in your environment by AWS Control Tower, according to the governance policies set by your organization. These AWS Regions are monitored to adhere to best practices and organizational policies. Your resources in these Regions are protected when you enable AWS Control Tower controls.
- **Not governed:** Regions that show **Not governed** status are not controlled or monitored by AWS Control Tower. These AWS Regions usually do not adhere to the same governance policies that AWS Control Tower enforces. You can create resources in these Regions, but those resources are not protected by AWS Control Tower controls.
- **Denied:** A denied Region is blocked specifically by AWS Control Tower. Within your AWS Control Tower environment, you cannot provision resources in these AWS Regions.

Pricing

No additional charge exists for using AWS Control Tower. You only pay for the AWS services enabled by AWS Control Tower, and the services you use in your landing zone. For example, you pay for Service Catalog for provisioning accounts with Account Factory, and AWS CloudTrail for events tracked in your landing zone. For information about the pricing and fees associated with AWS Control Tower, see AWS Control Tower pricing.

If you are running ephemeral workloads from accounts in AWS Control Tower, you may see an increase in costs associated with AWS Config. For details, see AWS Config pricing. Contact your AWS account representative for more specific information about managing these costs. To learn more about how AWS Config works with AWS Control Tower, see Monitor resource changes with AWS Config.

If you implement AWS CloudTrail trails outside of AWS Control Tower, you can use them with AWS Control Tower. However, you may incur duplicate charges, if you also opt in to trails managed by AWS Control Tower. We do not recommend setting up external trails, unless you have a specific requirement. If you choose to opt in during landing zone setup or update, AWS Control Tower sets up and activates an organization-level CloudTrail trail for you in the management account. For information about managing CloudTrail costs, see Managing CloudTrail costs.

Setting up

Before you use AWS Control Tower for the first time, follow the steps in this section to create an AWS account and protect your AWS Control Tower management account. For information on additional setup tasks specifically for AWS Control Tower, see Getting started with AWS Control Tower.

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS Control Tower. If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

Note your AWS account number, because you need it for other tasks.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see [Create a permission set](#) in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see [Add groups](#) in the *AWS IAM Identity Center User Guide*.

Security for your accounts

You can find additional guidance about how to set up best practices that protect the security of your AWS Control Tower accounts, in the AWS Organizations documentation.

- Best practices for the management account
- Best practices for member accounts

Next step

Getting started with AWS Control Tower

Getting started with AWS Control Tower

This getting started procedure is intended for AWS Control Tower administrators. Follow this procedure when you're ready to set up your landing zone using the AWS Control Tower console or APIs.

If you are an AWS customer currently, but new to AWS Control Tower, you may wish to review the section called Plan your AWS Control Tower landing zone, before you proceed.

Topics

AWS Control Tower quick start guide

If you are new to AWS, you can follow the steps in this section to get started quickly with AWS Control Tower. If you prefer to customize your AWS Control Tower environment right away, see Step 2. Configure and launch your landing zone .

> **Note**
>
> AWS Control Tower sets up paid services, such as AWS CloudTrail, AWS Config, Amazon CloudWatch, Amazon S3, and Amazon VPC. When used, these services may incur costs, as shown on the pricing page. The AWS management console shows you the usage of any paid services and the costs incurred. No additional costs are created by AWS Control Tower itself.

Before you begin

The most important decision to make before you begin the setup process is to *choose your home Region*. Your home Region is the AWS Region in which you'll run most of your workloads or store most of your data. It cannot be changed after you've set up your AWS Control Tower landing zone.

For more information about how to choose a home Region, see Administrative tips for landing zone setup .

> **Note**
>
> By default, AWS Control Tower chooses the Region in which your account is operating currently as your home Region. You can see your current Region in the upper right of your AWS management console screen.

The quick start procedure assumes that you'll accept the default values for the resources in your AWS Control Tower environment. Many of these choices can be changed later. A few one-time choices are listed in the section called Expectations for landing zone configuration .

If you've created a new AWS account, it automatically meets the required prerequisites for setting up AWS Control Tower. You can proceed through the steps that follow.

Quick start steps

1. Sign in to the AWS management console with your administrator user credentials.
2. Navigate to the **AWS Control Tower** console at https://console.aws.amazon.com/controltower.
3. Verify that you are working in your desired home Region.
4. Choose **Set up landing zone**.
5. Follow the instructions in the console, accepting all the default values. You will need to type in the email address for your account, a log archive account, and an audit account.
6. Confirm your choices and choose **Set up landing zone**.
7. AWS Control Tower takes about 30 minutes to set up all of the resources in your landing zone.

For a more detailed version of how to set up AWS Control Tower, including ways to customize your environment, read and follow the procedures in the next few topics.

> **Note**
>
> If you are a first-time customer and you encounter a setup issue, contact AWS Support for diagnostic assistance.

Prerequisite: Automated pre-launch checks for your management account

Before AWS Control Tower sets up the landing zone, it automatically runs a series of pre-launch checks in your account. There's no action required on your part for these checks, which ensure that your management account is ready for the changes that establish your landing zone. Here are the checks that AWS Control Tower runs before setting up a landing zone:

- The existing service limits for the AWS account must be sufficient for AWS Control Tower to launch. For more information, see Limitations and quotas in AWS Control Tower.
- The AWS account must be subscribed to the following AWS services:
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon SNS
  - Amazon Virtual Private Cloud (Amazon VPC)
  - AWS CloudFormation
  - AWS CloudTrail
  - Amazon CloudWatch
  - AWS Config
  - AWS Identity and Access Management (IAM)
  - AWS Lambda

> **Note**
>
> By default, all accounts are subscribed to these services.

Considerations for AWS IAM Identity Center (IAM Identity Center) customers

- If AWS IAM Identity Center (IAM Identity Center) is already set up, the AWS Control Tower home Region must be the same as the IAM Identity Center Region.
- IAM Identity Center can be installed only in the management account of an organization.

- Three options apply to your IAM Identity Center directory, based on the identity source you choose:
  - **IAM Identity Center User Store**: If AWS Control Tower is set up with IAM Identity Center, AWS Control Tower creates groups in the IAM Identity Center directory and provisions access to these groups, for the user you select, for member accounts.
  - **Active Directory**: If IAM Identity Center for AWS Control Tower is set up with Active Directory, AWS Control Tower does not manage the IAM Identity Center directory. It does not assign users or groups to new AWS accounts.
  - **External Identity Provider**: If IAM Identity Center for AWS Control Tower is set up with an external identity provider (IdP), AWS Control Tower creates groups in the IAM Identity Center directory and provisions access to these groups for the user you select for member accounts. You can specify an existing user from your external IdP in Account Factory during account creation, and AWS Control Tower gives this user access to the newly vended account when it synchronizes users of the same name between IAM Identity Center and the external IdP. You can also create groups in your external IdP to match the names of the default groups in AWS Control Tower. When you assign users to these groups, these users will have access to your enrolled accounts.

  For more information about working with IAM Identity Center and AWS Control Tower see Things to know about IAM Identity Center accounts and AWS Control Tower

Considerations for AWS Config and AWS CloudTrail customers

- The AWS account cannot have trusted access enabled in the organization management account for AWS Config. For information about how to disable trusted access, see the AWS Organizations documentation on how to enable or disable trusted access.
- If you have an existing AWS Config recorder, delivery channel, or aggregation setup in any existing accounts that you plan to enroll in AWS Control Tower, you must modify or remove these configurations before you start enrolling the accounts, after your landing zone is set up. This pre-check doesn't apply to the AWS Control Tower management account during landing zone launch. For more information, see Enroll accounts that have existing AWS Config resources.
- If you are running ephemeral workloads from accounts in AWS Control Tower, you may see an increase in costs associated with AWS Config. Contact your AWS account representative for more specific information about managing these costs.

- When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the AWS Control Tower organization. If you have an existing deployment of a CloudTrail trail in the account, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower. For information about organization-level trails and AWS Control Tower, see Pricing.

> **Note**
>
> When launching, AWS Security Token Service (STS) endpoints must be activated in the management account, for all Regions governed by AWS Control Tower. Otherwise, the launch may fail midway through the configuration process.

Get started with AWS Control Tower from the console

This getting started procedure is intended for AWS Control Tower administrators. Follow this procedure when you're ready to set up your landing zone using the AWS Control Tower console. From start to finish, it should take about half an hour. This procedure requires some prerequisites and three main steps.

If you are an AWS customer currently, but new to AWS Control Tower, you may wish to review the section called Plan your AWS Control Tower landing zone, before you proceed.

Topics

Expectations for landing zone configuration

The process of setting up your AWS Control Tower landing zone has multiple steps. Certain aspects of your AWS Control Tower landing zone are configurable. Other choices cannot be changed after setup.

Key items to configure during setup

- You can select your top-level OU names during setup, and you also can change OU names after you've set up your landing zone. By default, the top-level OUs are named **Security** and **Sandbox**. For more information, see Guidelines to set up a well-architected environment.
- During setup, you can select customized names for the shared accounts that AWS Control Tower creates, called **log archive** and **audit** by default, but you cannot change these names after setup. (This is a one-time selection.)
- During setup, you can optionally specify existing AWS accounts for AWS Control Tower to use as audit and log archive accounts. If you plan to specify existing AWS accounts, and if those accounts have existing AWS Config resources, you must delete the existing AWS Config resources before you can enroll the accounts into AWS Control Tower. (This is a one-time selection.)
- If you are setting up for the first time, or if you're upgrading to landing zone version 3.0, you can choose whether to allow AWS Control Tower to set up an organization-level AWS CloudTrail trail for your organization, or you can opt out of trails that are managed by AWS Control Tower and manage your own CloudTrail trails. You can opt into or opt out of organization-level trails that are managed by AWS Control Tower any time you update your landing zone.
- You can optionally set a customized retention policy for your Amazon S3 log bucket and log access bucket, when you set up or update your landing zone.
- You can optionally specify a previously-defined *blueprint* to use for provisioning customized member accounts from the AWS Control Tower console. You can customize accounts later if you do not have a blueprint available. See Customize accounts with Account Factory Customization (AFC).

Configuration choices that cannot be undone

- You cannot change your home Region after you've set up your landing zone.
- If you're provisioning Account Factory accounts with VPCs, VPC CIDRs can't be changed after they are created.

Step 1: Create your shared account email addresses

If you're setting up your landing zone in a new AWS account, see Setting up.

- To set up your landing zone with *new* shared accounts, AWS Control Tower requires two unique email addresses that aren't already associated with an AWS account. Each of these email addresses will serve as a collaborative inbox -- a shared email account -- intended for the various users in your enterprise that will do specific work related to AWS Control Tower.

- If you are setting up AWS Control Tower for the first time, and if you are bringing existing security and log archive accounts into AWS Control Tower, you can enter the current email addresses of the existing AWS accounts.

The email addresses are required for:

- **Audit account –** This account is for your team of users that need access to the audit information made available by AWS Control Tower. You can also use this account as the access point for third-party tools that will perform programmatic auditing of your environment to help you audit for compliance purposes.
- **Log archive account –** This account is for your team of users that need access to all the logging information for all of your enrolled accounts within registered OUs in your landing zone.

These accounts are set up in the **Security** OU when you create your landing zone. As a best practice, we recommend that when you perform actions in these accounts, you should use an IAM Identity Center user with the appropriately scoped permissions.

> **Note**
>
> If you specify existing AWS accounts as your **audit** and **log archive** accounts, the existing accounts must pass some pre-launch checks to ensure that no resources are in conflict with AWS Control Tower requirements. If these checks are not successful, your landing zone setup may not succeed. In particular, the accounts must not have existing AWS Config resources. For more information, see Considerations for bringing existing security or logging accounts.

For the sake of clarity, this *User Guide* always refers to the shared accounts by their default names: **log archive** and **audit**. As you read this document, remember to substitute the customized names you give to these accounts initially, if you choose to customize them. You can view your accounts with their customized names on the **Account details** page.

Note

We are changing our terminology regarding the default names of some AWS Control Tower organizational units (OUs) to align with the AWS multi-account strategy. You may notice some inconsistencies while we are making a transition to improve the clarity of these names. The Security OU was formerly called the Core OU. The Sandbox OU was formerly called the Custom OU.

Step 2. Configure and launch your landing zone

Before you launch your AWS Control Tower landing zone, determine the most appropriate home Region. For more information, see Administrative tips for landing zone setup.

Important

Changing your home Region after you have deployed your AWS Control Tower landing zone requires decommissioning as well as the assistance of AWS Support. This practice is not recommended.

Learn how to configure and launch your landing zone using the AWS CLI in Get started with AWS Control Tower using APIs.

To configure and launch your landing zone in the console, perform the following series of steps.

Prepare: Navigate to the AWS Control Tower console

1. Open a web browser, and navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. In the console, verify that you are working in your desired home Region for AWS Control Tower. Then choose **Set up your landing zone**.

Step 2a. Review and select your AWS Regions

Be sure you've correctly designated the AWS Region that you select for your home Region. After you've deployed AWS Control Tower, you can't change the home Region.

In this section of the setup process, you can add any additional AWS Regions that you require. You can add more Regions at a later time, if needed, and you can remove Regions from governance.

To select additional AWS Regions to govern

1. The panel shows you the current Region selections. Open the dropdown menu to see a list of additional Regions available for governance.
2. Check the box next to each Region to bring into governance by AWS Control Tower. Your home Region selection is not editable.

To deny access to certain Regions

To deny access to AWS resources and workloads in certain AWS Regions, select **Enabled** in the section for the Region deny control. By default, the setting for this control is **Not enabled**.

Step 2b. Configure your organizational units (OUs)

If you accept the default names of these OUs, there's no action you need to take for setup to continue. To change the names of the OUs, enter the new names directly in the form field.

- **Foundational OU** – AWS Control Tower relies upon a **Foundational OU** that is initially named the **Security OU**. You can change the name of this OU during initial setup and afterward, from the OU details page. This **Security OU** contains your two shared accounts, which by default are called the **log archive** account and the **audit** account.
- **Additional OU** – AWS Control Tower can set up one or more **Additional OUs** for you. We recommend that you provision at least one **Additional OU** in your landing zone, besides the **Security OU**. If this Additional OU is intended for development projects, we recommend that you name it the **Sandbox OU**, as given in the Guidelines to set up a well-architected environment. If you already have an existing OU in AWS Organizations, you may see the option to skip setting up an Additional OU in AWS Control Tower.

Step 2c. Configure your shared accounts, logging, and encryption

In this section of the setup process, the panel shows the default selections for the names of your shared AWS Control Tower accounts. These accounts are an essential part of your landing zone. **Do not move or delete these shared accounts**. You can choose customized names for the **audit** and **log archive** accounts during setup. Alternatively, you have a one-time option to specify existing AWS accounts as your shared accounts.

You must provide unique email addresses for your log archive and audit accounts, and you can verify the email address that you previously provided for your management account. Choose the **Edit** button to change the editable default values.

About the shared accounts

- **The management account –** The AWS Control Tower management account is part of the Root level. The management account allows for AWS Control Tower billing. The account also has administrator permissions for your landing zone. You cannot create separate accounts for billing and for administrator permissions in AWS Control Tower.

  The email address shown for the management account is not editable during this phase of setup. It is shown as a confirmation, so you can check that you're editing the correct management account, in case you have multiple accounts.
- **The two shared accounts –** You can choose customized names for these two accounts, or bring your own accounts, and you must supply a unique email address for each account, either new or existing. If you choose to have AWS Control Tower create new shared accounts for you, the email addresses must not already have associated AWS accounts.

To configure the shared accounts, fill in the requested information.

1. At the console, enter a name for the account initially called the **log archive** account. Many customers decide to keep the default name for this account.
2. Provide a unique email address for this account.
3. Enter a name for the account initially called the **audit** account. Many customers choose to call it the **Security** account.
4. Provide a unique email address for this account.

Optionally configure log retention

During this phase of setup, you can customize the log retention policy for Amazon S3 buckets that store your AWS CloudTrail logs in AWS Control Tower, in increments of days or years, up to a maximum of 15 years. If you choose not to customize your log retention, the default settings are one year for standard account logging and 10 years for access logging. This feature also is available when you update or reset your landing zone.

Optionally self-manage AWS account access

You can select whether AWS Control Tower sets up AWS account access with AWS Identity and Access Management (IAM), or whether to self-manage AWS account access—either with AWS IAM Identity Center users, roles, and permissions that you can set up and customize on your own, or with another method *such as an external IdP, either for direct account federation or federation to multiple accounts by means of IAM Identity Center*. You can change this selection later.

By default, AWS Control Tower sets up AWS IAM Identity Center for your landing zone, in alignment with best-practices guidance defined in [Organizing your AWS environment using multiple accounts](). Most customers choose the default. Alternative access methods are required sometimes, for regulatory compliance in specific industries or countries, or in AWS Regions where AWS IAM Identity Center is not available.

Selection of identity providers at the account level is not supported. This option applies only for the landing zone as a whole.

For more information, see [IAM Identity Center guidance]().

Optionally configure AWS CloudTrail trails

As a best practice, we recommend that you set up logging. If you wish to allow AWS Control Tower to set up an organization-level CloudTrail trail and manage it for you, choose **Opt in**. If you wish to manage logging with your own CloudTrail trails or a third-party logging tool, choose **Opt out**. Confirm your selection when requested to do so in the console. You can change your selection, and opt into, or opt out of, organization-level trails when you update your landing zone.

You can set up and manage your own CloudTrail trails at any time, including organization-level and account-level trails. If you set up duplicate CloudTrail trails, you may incur duplicate costs when CloudTrail events are logged.

Optionally configure AWS KMS keys

If you wish to encrypt and decrypt your resources with an AWS KMS encryption key, select the checkbox. If you have existing keys, you'll be able to select them from identifiers displayed in a dropdown menu. You can generate a new key by choosing **Create a key**. You can add or change a KMS key any time you update your landing zone.

When you select **Set up landing zone**, AWS Control Tower performs a pre-check to validate your KMS key. The key must meet these requirements:

- Enabled
- Symmetric
- Not a multi-Region key
- Has correct permissions added to the policy
- Key is in the management account

You may see an error banner if the key does not meet these requirements. In that case, choose another key or generate a key. Be sure to edit the key's permissions policy, as described in the next section.

Update the KMS key policy

Before you can update a KMS key policy, you must create a KMS key. For more information, see [Creating a key policy](#) in the *AWS Key Management Service Developer Guide*.

To use a KMS key with AWS Control Tower, you must update the default KMS key policy by adding the minimum required permissions for AWS Config and AWS CloudTrail. As a best practice, we recommend that you include the minimum required permissions in any policy. When updating a KMS key policy, you can add permissions as a group in a single JSON statement or line by line.

The procedure describes how to update the default KMS key policy in the AWS KMS console by adding policy statements that allow AWS Config and CloudTrail to use AWS KMS for encryption. The policy statements require that you include the following information:

- **`YOUR-MANAGEMENT-ACCOUNT-ID`** – the ID of the management account in which AWS Control Tower will be set up.
- **`YOUR-HOME-REGION`** – the home Region that you will select when setting up AWS Control Tower.
- **`YOUR-KMS-KEY-ID`** – the KMS key ID that will be used with the policy.

To update the KMS key policy

1. Open the AWS KMS console at https://console.aws.amazon.com/kms
2. From the navigation pane, choose **Customer managed keys**.
3. In the table, select the key that you want to edit.
4. In the **Key policy** tab, make sure that you can view the key policy. If you can't view the key policy, choose **Switch to policy view**.

5. Choose **Edit**, and update the default KMS key policy by adding the following policy statements for AWS Config and CloudTrail.

AWS Config policy statement

```
{
    "Sid": "Allow Config to use KMS for encryption",
    "Effect": "Allow",
    "Principal": {
        "Service": "config.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "arn:aws:kms:YOUR-HOME-REGION:YOUR-MANAGEMENT-ACCOUNT-ID:key/YOUR-
KMS-KEY-ID"
}
```

CloudTrail policy statement

```
{
    "Sid": "Allow CloudTrail to use KMS for encryption",
    "Effect": "Allow",
    "Principal": {
        "Service": "cloudtrail.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
    ],
    "Resource": "arn:aws:kms:YOUR-HOME-REGION:YOUR-MANAGEMENT-ACCOUNT-ID:key/YOUR-
KMS-KEY-ID",
    "Condition": {
        "StringEquals": {
            "aws:SourceArn": "arn:aws:cloudtrail:YOUR-HOME-REGION:YOUR-MANAGEMENT-
ACCOUNT-ID:trail/aws-controltower-BaselineCloudTrail"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:cloudtrail:arn": "arn:aws:cloudtrail:*:YOUR-
MANAGEMENT-ACCOUNT-ID:trail/*"
        }
```

```
    }
}
```

6. Choose **Save changes**.

Example KMS key policy

The following example policy shows what your KMS key policy might look like after you add the policy statements that grant AWS Config and CloudTrail the minimum required permissions. The example policy doesn't include your default KMS key policy.

```
{
    "Version": "2012-10-17",
    "Id": "CustomKMSPolicy",
    "Statement": [
        {
        ... YOUR-EXISTING-POLICIES ...
        },
        {
            "Sid": "Allow Config to use KMS for encryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "config.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:PARTITION:kms:YOUR-HOME-REGION:YOUR-MANAGEMENT-ACCOUNT-ID:key/YOUR-KMS-KEY-ID"
        },
        {
            "Sid": "Allow CloudTrail to use KMS for encryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
              ],
            "Resource": "arn:PARTITION:kms:YOUR-HOME-REGION:YOUR-MANAGEMENT-ACCOUNT-ID:key/YOUR-KMS-KEY-ID",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:PARTITION:cloudtrail:YOUR-HOME-REGION:YOUR-
MANAGEMENT-ACCOUNT-ID:trail/aws-controltower-BaselineCloudTrail"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:cloudtrail:arn":
 "arn:PARTITION:cloudtrail:*:YOUR-MANAGEMENT-ACCOUNT-ID:trail/*"
                }
            }
        }
    ]
}
```

To view other example policies, see the following pages:

- Granting encrypt permissions in the *AWS CloudTrail User Guide*.
- Required Permissions for the KMS Key When Using Service-Linked RolesS3 Bucket Delivery) in the *AWS Config Developer Guide*.

Protect against attackers

By adding certain conditions to your policies, you can help prevent a specific type of attack, known as a *confused deputy* attack, which occurs if an entity coerces a more-privileged entity to perform an action, such as with cross-service impersonation. For general information about policy conditions, also see Specifying conditions in a policy.

The AWS Key Management Service (AWS KMS) allows you to create multi-Region KMS keys and asymmetric keys; however, AWS Control Tower does not support multi-Region keys or asymmetric keys. AWS Control Tower performs a pre-check of your existing keys. You may see an error message if you select a multi-Region key or an asymmetric key. In that case, generate another key for use with AWS Control Tower resources.

For more information about AWS KMS, see the AWS KMS Developer Guide.

Note that customer data in AWS Control Tower is encrypted at rest, by default, using SSE-S3.

Optionally configure and create customized member accounts

When you follow the **Create account** workflow to add your member accounts, you can optionally specify a previously-defined *blueprint* to use for provisioning customized member accounts from the AWS Control Tower console. You can customize accounts later if you do not have a blueprint available. See Customize accounts with Account Factory Customization (AFC).

Step 3. Review and set up the landing zone

The next section in the setup shows you the permissions that AWS Control Tower requires for your landing zone. Choose a checkbox to expand each topic. You'll be asked to agree to these permissions, which may affect multiple accounts, and to agree to the overall **Terms of Service**.

To finalize

1. At the console, review the **Service permissions**, and when you're ready, choose **I understand the permissions AWS Control Tower will use to administer AWS resources and enforce rules on my behalf**.
2. To finalize your selections and initialize launch, choose **Set up landing zone**.

This series of steps starts the process of setting up your landing zone, which can take about thirty minutes to complete. During setup, AWS Control Tower creates your Root level, the Security OU, and the shared accounts. Other AWS resources are created, modified, or deleted.

> **Confirm SNS subscriptions**
>
> The email address you provided for the audit account will receive **AWS Notification – Subscription Confirmation** emails from every AWS Region supported by AWS Control Tower. To receive compliance emails in your audit account, you must choose the **Confirm subscription** link within each email from each AWS Region supported by AWS Control Tower.

Get started with AWS Control Tower using APIs

This getting started procedure is intended for AWS Control Tower administrators. This procedure requires some prerequisites and includes two main steps.

In this procedure, you will use APIs from AWS Control Tower and other AWS services to configure and launch a landing zone. These APIs allow you to create a AWS Control Tower environment programatically, either through the AWS CloudFormation console, or through the AWS CLI.

Before you launch your AWS Control Tower landing zone, perform these prerequisite tasks:

- Determine the most appropriate home Region. For more information, see Administrative tips for landing zone setup .
- Review Prerequisite: Automated pre-launch checks for your management account to learn about the automated pre-launch checks that make sure your management account is ready for changes that establish your landing zone.

Topics

Expectations for landing zone configuration with APIs

The process of setting up your AWS Control Tower landing zone has multiple steps. Certain aspects of your AWS Control Tower landing zone are configurable. Other choices cannot be changed after setup.

Key items to configure during setup

- You can select your Foundational OU names during setup, and you also can change OU names after you've set up your landing zone. By default, the Foundational OUs are named **Security** and **Sandbox**. For more information, see Guidelines to set up a well-architected environment.
- During setup, you can select customized names for the shared accounts that AWS Control Tower creates, called **log archive** and **audit** by default, but you cannot change these names after setup. (This is a one-time selection.)
- During setup with APIs, you *must* specify existing AWS accounts for AWS Control Tower to use as audit and log archive accounts. To specify existing AWS accounts, if those accounts have existing AWS Config resources, you must delete or modify the existing AWS Config resources before you can enroll the accounts into AWS Control Tower. (This is a one-time selection.)
- If you are setting up for the first time, or if you're upgrading to landing zone version 3.0, you can choose whether to allow AWS Control Tower to set up an organization-level AWS CloudTrail trail for your organization, or you can opt out of trails that are managed by AWS Control Tower and manage your own CloudTrail trails. You can opt into or opt out of organization-level trails that are managed by AWS Control Tower any time you update your landing zone.
- You can optionally set a customized retention policy for your Amazon S3 log bucket and log access bucket, when you set up or update your landing zone.

Configuration choices that cannot be undone

- You cannot change your home Region after you've set up your landing zone.
- If you're provisioning accounts with VPCs, VPC CIDRs can't be changed after they are created.

The next sections give the setup prerequisites and steps in detail, with explanations and caveats. For additional code examples, see Examples: Set up an AWS Control Tower landing zone with APIs only.

Step 1: Configure your landing zone

The process of setting up your AWS Control Tower landing zone has multiple steps. Certain aspects of your AWS Control Tower landing zone are configurable, but other choices cannot be changed after setup. To learn more about these important considerations prior to launching your landing zone, review Expectations for landing zone configuration .

Before using the AWS Control Tower landing zone APIs, you must first call APIs from other AWS services to configure your landing zone prior to launch. The process includes three main steps:

- creating a new AWS Organizations organization,
- setting up your shared account email addresses,
- and creating an IAM role or IAM Identity Center user with the required permissions to call the landing zone APIs.

Step 1. Create the organization that will contain your landing zone:

1. Call the AWS Organizations `CreateOrganization` API and enable all features to create the **Foundational OU**. AWS Control Tower initially names this the **Security OU**. This Security OU contains your two shared accounts, which by default are called the **log archive** account and the **audit** account.

   ```
   aws organizations create-organization --feature-set ALL
   ```

   AWS Control Tower can set up one or more **Additional OUs**. We recommend that you provision at least one Additional OU in your landing zone, besides the Security OU. If this Additional OU is intended for development projects, we recommend that you name it the **Sandbox OU**, as given in the AWS multi-account strategy for your AWS Control Tower landing zone.

Step 2. Provision shared accounts if needed:

To set up your landing zone, AWS Control Tower requires two email addresses. If you are using landing zone APIs to set up AWS Control Tower for the first time, you *must* use existing security and log archive AWS accounts. You can use the current email addresses of the existing AWS accounts. Each of these email addresses will serve as a collaborative inbox -- a shared email account -- intended for the various users in your enterprise that will do specific work related to AWS Control Tower.

To begin setting up a new landing zone, if you don't have existing AWS accounts, you can provision the security and log archive AWS accounts using AWS Organizations APIs.

1. Call the AWS Organizations `CreateAccount` API to create the **Log archive** account and **Audit** account in the **Security OU**.

```
aws organizations create-account --email mylog@example.com --account-name "Logging
 Account"
```

```
aws organizations create-account --email mysecurity@example.com --account-name
 "Security Account"
```

2. (Optional) Check the status of the `CreateAccount` operation using the AWS Organizations `DescribeAccount` API.

Step 3. Create the required service roles

Create the following IAM service roles in the `/service-role/` IAM path that enable AWS Control Tower to perform the API calls required to set up your landing zone:

- `AWSControlTowerAdmin`
- `AWSControlTowerCloudTrailRole`
- `AWSControlTowerStackSetRole`
- `AWSControlTowerConfigAggregatorRoleForOrganizations`

For more information about these roles and their policies, see Using identity-based policies (IAM policies) for AWS Control Tower.

To create an IAM role:

1. Create an IAM role with the necessary permissions to call all landing zone APIs. Alternatively, you can create an IAM Identity Center user and assign the necessary permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "backup:UpdateGlobalSettings",
                "controltower:CreateLandingZone",
                "controltower:UpdateLandingZone",
                "controltower:ResetLandingZone",
                "controltower:DeleteLandingZone",
```

```
                "controltower:GetLandingZoneOperation",
                "controltower:GetLandingZone",
                "controltower:ListLandingZones",
                "controltower:ListLandingZoneOperations",
                "controltower:ListTagsForResource",
                "controltower:TagResource",
                "controltower:UntagResource",
                "servicecatalog:*",
                "organizations:*",
                "organizations:RegisterDelegatedAdministrator",
                "organizations:EnableAWSServiceAccess",
                "organizations:DeregisterDelegatedAdministrator"
                "sso:*",
                "sso-directory:*",
                "logs:*",
                "cloudformation:*",
                "kms:*",
                "iam:GetRole",
                "iam:CreateRole",
                "iam:GetSAMLProvider",
                "iam:CreateSAMLProvider",
                "iam:CreateServiceLinkedRole",
                "iam:ListRolePolicies",
                "iam:PutRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:AttachRolePolicy",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Step 2: Launch your landing zone

The AWS Control Tower `CreateLandingZone` API requires a landing zone version and a *landing zone manifest file* as input parameters. You can use the AWS Control Tower landing zone manifest file to configure the following features:

- Optionally configure log retention

- Optionally self-manage AWS account access
- Optionally configure AWS CloudTrail trails
- Optionally configure AWS KMS keys

After compiling your manifest file, you're ready to create a new landing zone.

For more information about what's in the manifest file, see View the details of your landing zone manifest file.

For more information about landing zone schemas that apply to the landing zone manifest file, see Landing zone schemas.

> **Note**
>
> AWS Control Tower does not support the Region deny control when using APIs to configure and launch a landing zone. After successfully launching your landing zone using APIs, you can use the AWS Control Tower console to Configure the Region deny control.

1. Call the AWS Control Tower `CreateLandingZone` API. This API requires a landing zone version and a landing zone manifest file as input.

   ```
   aws controltower create-landing-zone --landing-zone-version 3.3 --manifest "file://
   LandingZoneManifest.json"
   ```

   For more detail about the contents of the *landing zone manifest file*, see View the details of your landing zone manifest file.

   The following example shows a **LandingZoneManifest.json** manifest, which includes settings for governed Regions and centralized logging:

   ```
   {
      "governedRegions": ["us-west-2","us-west-1"],
      "organizationStructure": {
          "security": {
              "name": "CORE"
          },
          "sandbox": {
              "name": "Sandbox"
   ```

```
        }
    },
    "centralizedLogging": {
         "accountId": "222222222222",
         "configurations": {
             "loggingBucket": {
                 "retentionDays": 60
             },
             "accessLoggingBucket": {
                 "retentionDays": 60
             },
             "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/
e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX"
         },
         "enabled": true
    },
    "securityRoles": {
         "accountId": "333333333333"
    },
    "accessManagement": {
         "enabled": true
    }
}
```

> **Note**
>
> As shown in the example, the **AccountId** for the `CentralizedLogging` and `SecurityRoles` accounts must be different.

The following example shows a **LandingZoneManifest.json** manifest file, which includes settings for backup and centralized logging:

```
{
        "landingZoneIdentifier": "LANDING ZONE ARN",
        "manifest": {
            "accessManagement": {
                "enabled": true
            },
            "securityRoles": {
                "accountId": "333333333333"
            },
```

```
            "backup": {
                "configurations": {
                    "centralBackup": {
                        "accountId": "CENTRAL BACKUP ACCOUNT ID"
                    },
                    "backupAdmin": {
                        "accountId": "BACKUP MANAGER ACCOUNT ID"
                    },
                    "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/
e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX"
                },
                "enabled": true
            },
            "governedRegions": [
                "us-west-1"
            ],
            "organizationStructure": {
                "sandbox": {
                    "name": "Sandbox"
                },
                "security": {
                    "name": "Security"
                }
            },
            "centralizedLogging": {
                "accountId": "222222222222",
                "configurations": {
                    "loggingBucket": {
                        "retentionDays": 365
                    },
                    "accessLoggingBucket": {
                        "retentionDays": 3650
                    }
                },
                "enabled": true
            }
        },
        "version": "3.3"
}
```

Output:

```
{
```

```
    "arn": "arn:aws:controltower:us-west-2:123456789012:landingzone/1A2B3C4D5E6F7G8H",
    "operationIdentifier": "55XXXXXX-e2XX-41XX-a7XX-446XXXXXXXXX"
}
```

2. Call the `GetLandingZoneOperation` API to check the status of the `CreateLandingZone` operation. The `GetLandingZoneOperation` API returns a **status** of `SUCCEEDED`, `FAILED`, or `IN_PROGRESS`.

```
aws controltower get-landing-zone-operation --operation-identifier "55XXXXXX-
eXXX-4XXX-aXXX-44XXXXXXXXXX"
```

Output:

```
{
    "operationDetails": {
        "operationType": "CREATE",
        "startTime": "Thu Nov 09 20:39:19 UTC 2023",
        "endTime": "Thu Nov 09 21:02:01 UTC 2023",
        "status": "SUCCEEDED"
    }
}
```

3. When the status returns as `SUCCEEDED`, you can call the `GetLandingZone` API to review the landing zone configuration.

```
aws controltower get-landing-zone --landing-zone-identifier "arn:aws:controltower:us-
west-2:123456789123:landingzone/1A2B3C4D5E6F7G8H"
```

Output:

```
{
    "landingZone": {
        "arn": "arn:aws:controltower:us-
west-2:123456789012:landingzone/1A2B3C4D5E6F7G8H",
        "driftStatus": {
            "status": "IN_SYNC"
        },
        "latestAvailableVersion": "3.3",
        "manifest": {
            "accessManagement": {
                "enabled": true
```

```
            },
            "securityRoles": {
                "accountId": "333333333333"
            },
            "governedRegions": [
                "us-west-1",
                "eu-west-3",
                "us-west-2"
            ],
            "organizationStructure": {
                "sandbox": {
                    "name": "Sandbox"
                },
                "security": {
                    "name": "Security"
                }
            },
            "centralizedLogging": {
                "accountId": "222222222222",
                "configurations": {
                    "loggingBucket": {
                        "retentionDays": 60
                    },
                    "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/
e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX",
                    "accessLoggingBucket": {
                        "retentionDays": 60
                    }
                },
                "enabled": true
            }
        },
        "status": "PROCESSING",
        "version": "3.3"
    }
}
```


Identify your landing zone

Calling `ListLandingZones` can help you determine if your account is already set up with AWS Control Tower. This API returns one landing zone identifier (ARN) across any **commercial** region, regardless of the landing zone's home region. Landing zone ARNs are regionally unique.

```
aws controltower list-landing-zones --region us-east-1
```

For opt-in regions, the `ListLandingZones` API only returns the landing zone identifier *if you call the API in the same region as the API's home region*. For example, if your landing zone is set up in af-south-1 and you call `ListLandingZones` *in af-south-1*, the API returns the landing zone identifier. If your landing zone is set up in af-south-1 and you call `ListLandingZones` *in ap-east-1*, the API **does not** return the landing zone identifier.

Output:

```
{
   "landingZones" [
         "arn": "arn:aws:controltower:us-
west-2:123456789123:landingzone/1A2B3C4D5E6F7G8H"
   ]
}
```

Update your landing zone

When a new landing zone version is available, or to make other updates to your landing zone configuration, you can call the `UpdateLandingZone` API and reference an updated landing zone manifest file. This API returns an `OperationIdentifier`, which you can then use when calling the `GetLandingZoneOperation` API to check the update operation's status.

To update the landing zone

1. Call the AWS Control Tower `UpdateLandingZone` API and refer to the updated **landing zone version** or your **updated landing zone manifest file**.

   ```
   aws controltower update-landing-zone --landing-zone-version 3.3 --landing-zone-
   identifier "arn:aws:controltower:us-west-2:123456789123:landingzone/1A2B3C4D5E6F7G8H"
    --manifest file://LandingZoneManifest.json
   ```

Example LandingZoneManifest.json file, with Regions and centralized logging:

```
{
   "governedRegions": ["us-west-2","us-west-1"],
   "organizationStructure": {
```

```
        "security": {
            "name": "Security"
        },
        "sandbox": {
            "name": "Sandbox"
        }
    },
    "centralizedLogging": {
         "accountId": "LOG ARCHIVE ACCOUNT ID",
         "configurations": {
             "loggingBucket": {
                 "retentionDays":2555
             },
             "accessLoggingBucket": {
                 "retentionDays": 2555
             },
             "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/
e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX"
         },
         "enabled": true
    },
    "securityRoles": {
         "accountId": "SECURITY ACCOUNT ID"
    },
    "accessManagement": {
         "enabled": true
    }
}
```

Output:

```
{
   "operationIdentifier": "55XXXXXX-e2XX-41XX-a7XX-446XXXXXXXXX"
}
```

Optionally Re-register OU to update accounts

For registered AWS Control Tower OUs with fewer than 1000 accounts, you can use the AWS Control Tower console access the **OU page** in the dashboard and select **Re-register OU** to update the accounts in that OU.

Reset the landing zone to resolve drift

When you create your landing zone, the landing zone and all the organizational units (OUs), accounts, and resources are compliant with the governance rules enforced by your chosen controls. As you and your organization members use the landing zone, changes in this compliance status may occur. These changes are called *drift*.

To identify if your landing zone is in drift, you can call the `GetLandingZone` API. This API returns the landing zone's **drift status** of `DRIFTED` or `IN_SYNC`.

To resolve drift within your landing zone you can use the `ResetLandingZone` API to reset the landing zone back to its original configuration. For example, AWS Control Tower enables IAM Identity Center by default to help you manage your AWS accounts-- but if you configure your original landing zone parameters with IAM Identity Center disabled, calling `ResetLandingZone` maintains that disabled IAM Identity Center configuration.

You can only use the `ResetLandingZone` API if you are using the latest available landing zone version. You can call the `GetLandingZone` API and compare your landing zone version with the **latest available version**. If necessary, you can Update your landing zone so your landing zone uses the latest available version. In these examples, we are using version 3.3 as the latest version.

1. Call the `GetLandingZone` API. If the API returns a **drift status** of `DRIFTED`, your landing zone is in drift.
2. Call the `ResetLandingZone` API to reset your landing zone to its original configuration.

```
aws controltower reset-landing-zone --landing-zone-identifier
 "arn:aws:controltower:us-west-2:123456789123:landingzone/1A2B3C4D5E6F7G8H"
```

Output:

```
{
   "operationIdentifier": "55XXXXXX-e2XX-41XX-a7XX-446XXXXXXXXX"
}
```

Note

Resetting the landing zone does not update the landing zone version. Review Update your landing zone for details about updating the landing zone version.

View the details of your landing zone manifest file

The AWS Control Tower landing zone manifest file is a text file that describes your AWS Control Tower resources. The following sections show detailed definitions of entries in the landing zone manifest file.

To see a full landing zone schema example, see Landing zone schemas.

governedRegions – Regions to place under governance

- **Type:** List of strings
- **Required:** No
- **Example:**

```
"governedRegions": ["us-west-2","us-west-1"]
```

organizationStructure – Select the names of security and sandbox OUs to be created in your organization

- **Type:** Object
- **Required:** Yes
- **Properties:**
- **Example:**
  - `security` - an object with one required property, `name`, which takes a `String`
  - `sandbox` - an object with one required property, `name`, which takes a `String`

```
"organizationStructure": {
        "security": {
            "name": "CORE"
        },
        "sandbox": {
            "name": "Sandbox"
        }
    }
```

centralizedLogging – Configuration for AWS CloudTrail

- **Type:** Object
- **Required:** Yes
- **Properties:**
  - *accountId* - a `String` the represents the AWS account into which the logging resource should be deployed
  - *configurations* - an `Object` with three properties
    - `loggingBucket` - an object with one property, `retentionDays`, which takes a `Number`
    - `accessLoggingBucket` - an object with one property, `retentionDays`, which takes a `Number`
    - `kmsKeyArn` - an optional `String`
  - *enabled* - an optional `Boolean`
- **Example:**

```
"centralizedLogging": {
        "accountId": "222222222222",
        "configurations": {
            "loggingBucket": {
                "retentionDays": 60
            },
            "accessLoggingBucket": {
                "retentionDays": 60
            },
            "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/
e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX"
        },
        "enabled": true
   }
```

securityRoles – Choose where to deploy the logging resource

- **Type:** Object
- **Required:** Yes
- **Properties:** *accountId* - a `String` that represents the AWS account into which the logging resource should be deployed
- **Example:**

```
"securityRoles": {
        "accountId": "333333333333"
   }
```

accessManagement – Choose whthether to enable access management

- **Type:** Object
- **Required:** No
- **Properties:** *enabled* - a Boolean
- **Example:**

```
"accessManagement": {
        "enabled": true
   }
```

backup – Configuration for AWS Backup with AWS Control Tower

- **Type:** Object
- **Required:** No
- **Properties:**
  - *configurations* - an `Object` with three properties
    - `centralBackup` - an object with one property, `accountId`, which takes a `String`
    - `backupAdmin` - an object with one property, `accountId`, which takes a `String`
    - `kmsKeyArn` - an optional `String`
  - *enabled* - a `Boolean`
- **Example:**

```
"backup": {
    "configurations": {
        "centralBackup": {
            "accountId": "CENTRAL BACKUP ACCOUNT ID"
        },
        "backupAdmin": {
            "accountId": "BACKUP MANAGER ACCOUNT ID"
        },
```

```
        "kmsKeyArn": "arn:aws:kms:us-west-1:123456789123:key/e84XXXXX-6bXX-49XX-9eXX-ecfXXXXXXXXX"
    },
    "enabled": true
}
```

View the status of your landing zone operations

The `ListLandingZoneOperations` API allows you to view the status of AWS Control Tower operations that perform actions on your landing zone.

For more information about this API operation, see ListLandingZoneOperations.

ListLandingZoneOperations

Example input and output for `ListLandingZoneOperations`.

This example shows how to call the API with no parameters.

```
aws controltower --region us-east-1 list-landing-zone-operations

{
    "landingZoneOperations": [
        {
            "operationIdentifier": "873fe98d-1ecc-4154-b593-86e4a95ebfXX",
            "operationType": "CREATE",
            "status": "FAILED"
        },
        {
            "operationIdentifier": "0016d43d-a307-4ad8-a2a2-b427b8eb1cXX",
            "operationType": "DELETE",
            "status": "SUCCEEDED"
        },
        {
            "operationIdentifier": "002b8b5a-6bb7-4c40-89cd-5822a73d13XX",
            "operationType": "CREATE",
            "status": "SUCCEEDED"
        },
        {
            "operationIdentifier": "008886a0-f7a2-4df3-90e8-6e9f936507XX",
            "operationType": "CREATE",
            "status": "FAILED"
```

```
        }
    ]
}
```

This example shows how to call the API and specify the maximum number of results.

```
aws controltower --region us-east-1 list-landing-zone-operations --max-results 1

{
    "landingZoneOperations": [
        {
            "operationIdentifier": "873fe98d-1ecc-4154-b593-86e4a95ebfXX",
            "operationType": "CREATE",
            "status": "FAILED"
        }
    ],
    "nextToken": "AAMAATFMzwP0QysYY8npWgstfcHGQBj-
XCC18ISyd9mkQmzLR7ZFMket4F0aWv8tUTtnsTWOnfblUp_Q9U-
nX9_6lEsLHs0RlhceDKskHr0_3fm8KdPTa6ofxMt5SPw8WF7-Jsvw2rJVvhj4DHDipo-y1HVK_eZ__Z3-
OzInm403cIHxhbjGPgqCX6FeKr8lwgTDKOejkLYZ9w7J5aqPAKLfVP8KKNda5g0VfMj1wdl4J2nwnHI-
UuCTIZ5nUEgXgUHaFq6Ma1pLDfGefZQJn5HmDhhgd5yvqzSRH1BtrHpdV_N1EVP8u3JJr3eWQHe9jNB02lihD4Mdcbm3SJg
VXRwTUIBInrit4HslNtPE8-IC1gxCjGoYPGtuWBPumK-pUPE="
}
```

This example shows how to call the API and obtain a paginated result with `nextToken`.

```
aws controltower --region us-east-1 list-landing-zone-operations --next-token
 AAMAATFMzwP0QysYY8npWgstfcHGQBj-XCC18ISyd9mkQmzLR7ZFMket4F0aWv8tUTtnsTWOnfblUp_Q9U-
nX9_6lEsLHs0RlhceDKskHr0_3fm8KdPTa6ofxMt5SPw8WF7-Jsvw2rJVvhj4DHDipo-y1HVK_eZ__Z3-
OzInm403cIHxhbjGPgqCX6FeKr8lwgTDKOejkLYZ9w7J5aqPAKLfVP8KKNda5g0VfMj1wdl4J2nwnHI-
UuCTIZ5nUEgXgUHaFq6Ma1pLDfGefZQJn5HmDhhgd5yvqzSRH1BtrHpdV_N1EVP8u3JJr3eWQHe9jNB02lihD4Mdcbm3SJg
VXRwTUIBInrit4HslNtPE8-IC1gxCjGoYPGtuWBPumK-pUPE=

{
    "landingZoneOperations": [
        {
            "operationIdentifier": "0016d43d-a307-4ad8-a2a2-b427b8eb1cXX",
            "operationType": "DELETE",
            "status": "SUCCEEDED"
        },
        {
            "operationIdentifier": "002b8b5a-6bb7-4c40-89cd-5822a73d13XX",
            "operationType": "CREATE",
```

```
            "status": "SUCCEEDED"
        },
        {
            "operationIdentifier": "008886a0-f7a2-4df3-90e8-6e9f936507XX",
            "operationType": "CREATE",
            "status": "FAILED"
        }
    ]
}
```

This example shows how to call the API with a filter.

```
aws controltower --region us-east-1 list-landing-zone-operations --filter '{"types":
["CREATE"],"statuses":["FAILED"]}'

{
    "landingZoneOperations": [
        {
            "operationIdentifier": "873fe98d-1ecc-4154-b593-86e4a95ebfXX",
            "operationType": "CREATE",
            "status": "FAILED"
        },
        {
            "operationIdentifier": "008886a0-f7a2-4df3-90e8-6e9f936507XX",
            "operationType": "CREATE",
            "status": "FAILED"
        }
    ]
}
```

Examples: Set up an AWS Control Tower landing zone with APIs only

This walkthrough of examples is a companion document. For explanations, caveats, and more information, see Getting started with AWS Control Tower using APIs.

Prerequisites

Before creating an AWS Control Tower landing zone, you must create an organization, two shared accounts, and some IAM roles. This walkthrough tutorial includes these steps, with example CLI commands and output.

Step 1. Create the organization and two required accounts.

```
aws organizations create-organization --feature-set ALL
aws organizations create-account --email example+log@example.com --account-name "Log
 archive account"
aws organizations create-account --email example+aud@example.com --account-name "Audit
 account"
```

Step 2. Create the required IAM roles.

AWSControlTowerAdmin

```
cat <<EOF >controltower_trust.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "controltower.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
aws iam create-role --role-name AWSControlTowerAdmin --path /service-role/ --assume-
role-policy-document file://controltower_trust.json
cat <<EOF >ct_admin_role_policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*"
        }
    ]
}
EOF
aws iam put-role-policy --role-name AWSControlTowerAdmin --policy-name
 AWSControlTowerAdminPolicy --policy-document file://ct_admin_role_policy.json
```

```
aws iam attach-role-policy --role-name AWSControlTowerAdmin --policy-arn
 arn:aws:iam::aws:policy/service-role/AWSControlTowerServiceRolePolicy
```

AWSControlTowerCloudTrailRole

```
cat <<EOF >cloudtrail_trust.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
aws iam create-role --role-name AWSControlTowerCloudTrailRole --path /service-role/ --
assume-role-policy-document file://cloudtrail_trust.json
cat <<EOF >cloudtrail_role_policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "logs:CreateLogStream",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/CloudTrailLogs:*",
            "Effect": "Allow"
        },
        {
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/CloudTrailLogs:*",
            "Effect": "Allow"
        }
    ]
}
EOF
aws iam put-role-policy --role-name AWSControlTowerCloudTrailRole --
policy-name AWSControlTowerCloudTrailRolePolicy --policy-document file://
cloudtrail_role_policy.json
```

AWSControlTowerStackSetRole

```
cat <<EOF >cloudformation_trust.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
aws iam create-role --role-name AWSControlTowerStackSetRole --path /service-role/ --
assume-role-policy-document file://cloudformation_trust.json
cat <<EOF >stackset_role_policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWSControlTowerExecution"
            ],
            "Effect": "Allow"
        }
    ]
}
EOF
aws iam put-role-policy --role-name AWSControlTowerStackSetRole --policy-name
 AWSControlTowerStackSetRolePolicy --policy-document file://stackset_role_policy.json
```

AWSControlTowerConfigAggregatorRoleForOrganizations

```
cat <<EOF >config_trust.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "config.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
aws iam create-role --role-name AWSControlTowerConfigAggregatorRoleForOrganizations --
path /service-role/ --assume-role-policy-document file://config_trust.json
aws iam attach-role-policy --role-name
 AWSControlTowerConfigAggregatorRoleForOrganizations --policy-arn
 arn:aws:iam::aws:policy/service-role/AWSConfigRoleForOrganizations
```

Step 3. Get account IDs and generate the landing zone manifest file.

The first two commands in the following example store the account IDs for the accounts you created in **Step 1** into variables. These variables then help generate the landing zone manifest file.

```
sec_account_id=$(aws organizations list-accounts | jq -r '.Accounts[] | select(.Name ==
 "Audit account") | .Id')
log_account_id=$(aws organizations list-accounts | jq -r '.Accounts[] | select(.Name ==
 "Log archive account") | .Id')

cat <<EOF >landing_zone_manifest.json
{
   "governedRegions": ["us-west-1", "us-west-2"],
   "organizationStructure": {
       "security": {
           "name": "Security"
       },
       "sandbox": {
           "name": "Sandbox"
       }
   },
   "centralizedLogging": {
        "accountId": "$log_account_id",
        "configurations": {
            "loggingBucket": {
                "retentionDays": 60
            },
            "accessLoggingBucket": {
                "retentionDays": 60
```

```
            }
        },
        "enabled": true
    },
    "securityRoles": {
        "accountId": "$sec_account_id"
    },
    "accessManagement": {
        "enabled": true
    }
}
EOF
```

Step 4. Create the landing zone with the latest version.

You must set up the landing zone with the manifest file and the latest version. This example shows version 3.3.

```
aws --region us-west-1 controltower create-landing-zone --manifest file://
landing_zone_manifest.json --landing-zone-version 3.3
```

The output will contain an **arn** and an **operationIdentifier**, as shown in the example that follows.

```
{
    "arn": "arn:aws:controltower:us-west-1:0123456789012:landingzone/4B3H0ULNUOL2AXXX",
    "operationIdentifier": "16bb47f7-b7a2-4d90-bc71-7df4ca1201xx"
}
```

Step 5. (Optional) Track the status of your landing zone creation operation, by setting up a loop.

To track status, use the **operationIdentifier** from the previous `create-landing-zone` command's output.

```
aws --region us-west-1 controltower get-landing-zone-operation --operation-identifier
 16bb47f7-b7a2-4d90-bc71-7df4ca1201xx
```

Sample status output:

```
{
    "operationDetails": {
```

```
        "operationType": "CREATE",
        "startTime": "2024-02-28T21:49:31Z",
        "status": "IN_PROGRESS"
    }
}
```

You can use the following example script to help you set up a loop, which reports the operation's status over and over, like a log file. Then you don't need to keep entering the command.

```
while true; do echo "$(date) $(aws --region us-west-1 controltower get-landing-
zone-operation --operation-identifier 16bb47f7-b7a2-4d90-bc71-7df4ca1201xx | jq -
r .operationDetails.status)"; sleep 15; done
```

To show detailed information about your landing zone

Step 1. Find the ARN of the landing zone

```
aws --region us-west-1 controltower list-landing-zones
```

Output will include the identifier of the landing zone, as shown in the following example of output.

```
{
    "landingZones": [
        {
            "arn": "arn:aws:controltower:us-
west-1:123456789012:landingzone/4B3H0ULNUOL2AXXX"
        }
    ]
}
```

Step 2. Get the information

```
aws --region us-west-1 controltower get-landing-zone --landing-zone-identifier
 arn:aws:controltower:us-west-1:123456789012:landingzone/4B3H0ULNUOL2AXXX
```

Here's an example of the kind of output you may see:

```
{
    "landingZone": {
        "arn": "arn:aws:controltower:us-
west-1:123456789012:landingzone/4B3H0ULNUOL2AXXX",
```

```
        "driftStatus": {
            "status": "IN_SYNC"
        },
        "latestAvailableVersion": "3.3",
        "manifest": {
            "accessManagement": {
                "enabled": true
            },
            "securityRoles": {
                "accountId": "9750XXXX4444"
            },
            "governedRegions": [
                "us-west-1",
                "us-west-2"
            ],
            "organizationStructure": {
                "sandbox": {
                    "name": "Sandbox"
                },
                "security": {
                    "name": "Security"
                }
            },
            "centralizedLogging": {
                "accountId": "012345678901",
                "configurations": {
                    "loggingBucket": {
                        "retentionDays": 60
                    },
                    "accessLoggingBucket": {
                        "retentionDays": 60
                    }
                },
                "enabled": true
            }
        },
        "status": "ACTIVE",
        "version": "3.3"
    }
}
```

Step 6. (Optional) Call the `ListLandingZoneOperations` API to view the status of any operations that change your landing zone.

To track the status of any landing zone operation, you can call the ListLandingZoneOperations API.

Landing zone schemas

A landing zone is an AWS resource, which is created by means of schemas. Each AWS Control Tower landing zone version has a unique schema.

The schemas for AWS Control Tower landing zones, version 3.1 and newer, are published in this reference section, to assist you in choosing a compatible version.

Note

A known issue regarding *unneccessary access logging* is present in landing zone version 3.0. The issue is addressed in landing zone version 3.1. For more information about the changes, see AWS Control Tower landing zone version 3.1.

Landing zone 3.3 schema

```
{
    "type": "object",
    "required": [
        "centralizedLogging",
        "organizationStructure",
        "securityRoles"
    ],
    "properties": {
        "accessManagement": {
            "$ref": "#/definitions/AccessManagement"
        },
        "backup": {
            "$ref": "#/definitions/Backup"
        },
        "centralizedLogging": {
            "$ref": "#/definitions/CentralizedLogging"
        },
        "governedRegions": {
            "type": "array",
            "items": {
                "type": "string",
                "maxLength": 24,
                "minLength": 1,
```

```
                "pattern": "^[a-z]{2}-[a-z\\-]*-[0-9]{1}$",
                "additionalProperties": false
            },
            "additionalProperties": false
        },
        "organizationStructure": {
            "$ref": "#/definitions/OrganizationStructure"
        },
        "securityRoles": {
            "$ref": "#/definitions/SecurityRoles"
        }
    },
    "additionalProperties": false,
    "definitions": {
        "AccessManagement": {
            "type": "object",
            "required": [
                "enabled"
            ],
            "properties": {
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
            "additionalProperties": false
        },
        "Backup": {
            "type": "object",
            "properties": {
                "configurations": {
                    "$ref": "#/definitions/BackupConfigurations"
                },
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": false
                }
            },
            "additionalProperties": false,
            "if": {
                "properties": {
                    "enabled": {
```

```
                        "const": true
                    }
                }
            },
            "then": {
                "required": [
                    "configurations"
                ]
            }
        },
        "BackupAdminConfigurations": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "BackupConfigurations": {
            "type": "object",
            "required": [
                "backupAdmin",
                "centralBackup",
                "kmsKeyArn"
            ],
            "properties": {
                "backupAdmin": {
                    "$ref": "#/definitions/BackupAdminConfigurations"
                },
                "centralBackup": {
                    "$ref": "#/definitions/CentralBackupConfigurations"
                },
                "kmsKeyArn": {
                    "type": "string",
                    "maxLength": 2048,
                    "minLength": 1,
```

```
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralBackupConfigurations": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralizedLogging": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                },
                "configurations": {
                    "$ref": "#/definitions/LoggingConfigurations"
                },
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
            "additionalProperties": false
```

```
        },
        "LoggingConfigurations": {
            "type": "object",
            "properties": {
                "accessLoggingBucket": {
                    "$ref": "#/definitions/S3BucketConfiguration"
                },
                "kmsKeyArn": {
                    "type": "string",
                    "maxLength": 2048,
                    "minLength": 1,
                    "additionalProperties": false
                },
                "loggingBucket": {
                    "$ref": "#/definitions/S3BucketConfiguration"
                }
            },
            "additionalProperties": false
        },
        "OrganizationalUnit": {
            "type": "object",
            "required": [
                "name"
            ],
            "properties": {
                "name": {
                    "type": "string",
                    "maxLength": 120,
                    "minLength": 1,
                    "pattern": "^[\\s\\S]*$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "OrganizationStructure": {
            "type": "object",
            "required": [
                "security"
            ],
            "properties": {
                "sandbox": {
                    "$ref": "#/definitions/OrganizationalUnit"
                },
```

```
                "security": {
                    "$ref": "#/definitions/OrganizationalUnit"
                }
            },
            "additionalProperties": false
        },
        "S3BucketConfiguration": {
            "type": "object",
            "properties": {
                "retentionDays": {
                    "type": "number",
                    "minimum": 1,
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "SecurityRoles": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        }
    }
}
```

Landing zone 3.2 schema

```
{
    "type": "object",
    "required": [
        "centralizedLogging",
        "organizationStructure",
```

```
        "securityRoles"
    ],
    "properties": {
        "accessManagement": {
            "$ref": "#/definitions/AccessManagement"
        },
        "backup": {
            "$ref": "#/definitions/Backup"
        },
        "centralizedLogging": {
            "$ref": "#/definitions/CentralizedLogging"
        },
        "governedRegions": {
            "type": "array",
            "items": {
                "type": "string",
                "maxLength": 24,
                "minLength": 1,
                "pattern": "^[a-z]{2}-[a-z\\-]*-[0-9]{1}$",
                "additionalProperties": false
            },
            "additionalProperties": false
        },
        "organizationStructure": {
            "$ref": "#/definitions/OrganizationStructure"
        },
        "securityRoles": {
            "$ref": "#/definitions/SecurityRoles"
        }
    },
    "additionalProperties": false,
    "definitions": {
        "AccessManagement": {
            "type": "object",
            "required": [
                "enabled"
            ],
            "properties": {
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
```

```
                "additionalProperties": false
            },
            "Backup": {
                "type": "object",
                "properties": {
                    "configurations": {
                        "$ref": "#/definitions/BackupConfigurations"
                    },
                    "enabled": {
                        "type": "boolean",
                        "additionalProperties": false,
                        "default": false
                    }
                },
                "additionalProperties": false,
                "if": {
                    "properties": {
                        "enabled": {
                            "const": true
                        }
                    }
                },
                "then": {
                    "required": [
                        "configurations"
                    ]
                }
            },
            "BackupAdminConfigurations": {
                "type": "object",
                "required": [
                    "accountId"
                ],
                "properties": {
                    "accountId": {
                        "type": "string",
                        "maxLength": 12,
                        "minLength": 12,
                        "pattern": "^\\d{12}$",
                        "additionalProperties": false
                    }
                },
                "additionalProperties": false
            },
```

```
        "BackupConfigurations": {
            "type": "object",
            "required": [
                "backupAdmin",
                "centralBackup",
                "kmsKeyArn"
            ],
            "properties": {
                "backupAdmin": {
                    "$ref": "#/definitions/BackupAdminConfigurations"
                },
                "centralBackup": {
                    "$ref": "#/definitions/CentralBackupConfigurations"
                },
                "kmsKeyArn": {
                    "type": "string",
                    "maxLength": 2048,
                    "minLength": 1,
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralBackupConfigurations": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralizedLogging": {
            "type": "object",
            "required": [
                "accountId"
            ],
```

```
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                },
                "configurations": {
                    "$ref": "#/definitions/LoggingConfigurations"
                },
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
            "additionalProperties": false
        },
        "LoggingConfigurations": {
            "type": "object",
            "properties": {
                "accessLoggingBucket": {
                    "$ref": "#/definitions/S3BucketConfiguration"
                },
                "kmsKeyArn": {
                    "type": "string",
                    "maxLength": 2048,
                    "minLength": 1,
                    "additionalProperties": false
                },
                "loggingBucket": {
                    "$ref": "#/definitions/S3BucketConfiguration"
                }
            },
            "additionalProperties": false
        },
        "OrganizationalUnit": {
            "type": "object",
            "required": [
                "name"
            ],
            "properties": {
                "name": {
```

```
                    "type": "string",
                    "maxLength": 120,
                    "minLength": 1,
                    "pattern": "^[\\s\\S]*$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "OrganizationStructure": {
            "type": "object",
            "required": [
                "security"
            ],
            "properties": {
                "sandbox": {
                    "$ref": "#/definitions/OrganizationalUnit"
                },
                "security": {
                    "$ref": "#/definitions/OrganizationalUnit"
                }
            },
            "additionalProperties": false
        },
        "S3BucketConfiguration": {
            "type": "object",
            "properties": {
                "retentionDays": {
                    "type": "number",
                    "minimum": 1,
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "SecurityRoles": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
```

```
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        }
    }
}
```

Landing zone 3.1 schema

```
{
    "type": "object",
    "required": [
        "centralizedLogging",
        "organizationStructure",
        "securityRoles"
    ],
    "properties": {
        "accessManagement": {
            "$ref": "#/definitions/AccessManagement"
        },
        "backup": {
            "$ref": "#/definitions/Backup"
        },
        "centralizedLogging": {
            "$ref": "#/definitions/CentralizedLogging"
        },
        "governedRegions": {
            "type": "array",
            "items": {
                "type": "string",
                "maxLength": 24,
                "minLength": 1,
                "pattern": "^[a-z]{2}-[a-z\\-]*-[0-9]{1}$",
                "additionalProperties": false
            },
            "additionalProperties": false
        },
        "organizationStructure": {
            "$ref": "#/definitions/OrganizationStructure"
        },
```

```
        "securityRoles": {
            "$ref": "#/definitions/SecurityRoles"
        }
    },
    "additionalProperties": false,
    "definitions": {
        "AccessManagement": {
            "type": "object",
            "required": [
                "enabled"
            ],
            "properties": {
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
            "additionalProperties": false
        },
        "Backup": {
            "type": "object",
            "properties": {
                "configurations": {
                    "$ref": "#/definitions/BackupConfigurations"
                },
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": false
                }
            },
            "additionalProperties": false,
            "if": {
                "properties": {
                    "enabled": {
                        "const": true
                    }
                }
            },
            "then": {
                "required": [
                    "configurations"
                ]
```

```
            }
        },
        "BackupAdminConfigurations": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "BackupConfigurations": {
            "type": "object",
            "required": [
                "backupAdmin",
                "centralBackup",
                "kmsKeyArn"
            ],
            "properties": {
                "backupAdmin": {
                    "$ref": "#/definitions/BackupAdminConfigurations"
                },
                "centralBackup": {
                    "$ref": "#/definitions/CentralBackupConfigurations"
                },
                "kmsKeyArn": {
                    "type": "string",
                    "maxLength": 2048,
                    "minLength": 1,
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralBackupConfigurations": {
            "type": "object",
            "required": [
```

```
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                }
            },
            "additionalProperties": false
        },
        "CentralizedLogging": {
            "type": "object",
            "required": [
                "accountId"
            ],
            "properties": {
                "accountId": {
                    "type": "string",
                    "maxLength": 12,
                    "minLength": 12,
                    "pattern": "^\\d{12}$",
                    "additionalProperties": false
                },
                "configurations": {
                    "$ref": "#/definitions/LoggingConfigurations"
                },
                "enabled": {
                    "type": "boolean",
                    "additionalProperties": false,
                    "default": true
                }
            },
            "additionalProperties": false
        },
        "LoggingConfigurations": {
            "type": "object",
            "properties": {
                "accessLoggingBucket": {
                    "$ref": "#/definitions/S3BucketConfiguration"
                },
                "kmsKeyArn": {
```

```
                "type": "string",
                "maxLength": 2048,
                "minLength": 1,
                "additionalProperties": false
            },
            "loggingBucket": {
                "$ref": "#/definitions/S3BucketConfiguration"
            }
        },
        "additionalProperties": false
    },
    "OrganizationalUnit": {
        "type": "object",
        "required": [
            "name"
        ],
        "properties": {
            "name": {
                "type": "string",
                "maxLength": 120,
                "minLength": 1,
                "pattern": "^[\\s\\S]*$",
                "additionalProperties": false
            }
        },
        "additionalProperties": false
    },
    "OrganizationStructure": {
        "type": "object",
        "required": [
            "security"
        ],
        "properties": {
            "sandbox": {
                "$ref": "#/definitions/OrganizationalUnit"
            },
            "security": {
                "$ref": "#/definitions/OrganizationalUnit"
            }
        },
        "additionalProperties": false
    },
    "S3BucketConfiguration": {
        "type": "object",
```

```
                "properties": {
                    "retentionDays": {
                        "type": "number",
                        "minimum": 1,
                        "additionalProperties": false
                    }
                },
                "additionalProperties": false
            },
            "SecurityRoles": {
                "type": "object",
                "required": [
                    "accountId"
                ],
                "properties": {
                    "accountId": {
                        "type": "string",
                        "maxLength": 12,
                        "minLength": 12,
                        "pattern": "^\\d{12}$",
                        "additionalProperties": false
                    }
                },
                "additionalProperties": false
            }
        }
}
```

Launch a landing zone using AWS CloudFormation

You can configure and launch a landing zone with AWS CloudFormation either through the AWS CloudFormation console, or through the AWS CLI. This section provides instructions and examples to launch a landing zone using APIs through AWS CloudFormation.

Topics

- Prerequisites for launching a landing zone using AWS CloudFormation
- Create a new landing zone using AWS CloudFormation
- Manage an existing landing zone using AWS CloudFormation

Prerequisites for launching a landing zone using AWS CloudFormation

1. From the AWS CLI, use the AWS Organizations `CreateOrganization` API to create an organization and enable all features.

   For more detailed instructions, review [Step 1: Configure your landing zone](#).

2. From the AWS CloudFormation console or using the AWS CLI, deploy a AWS CloudFormation template that creates the following resources in the management account:

   - Log Archive account (sometimes called the "Logging" account)
   - Audit account (sometimes called the "Security" account)
   - The **AWSControlTowerAdmin**, **AWSControlTowerCloudTrailRole**, **AWSControlTowerConfigAggregatorRoleForOrganizations**, and **AWSControlTowerStackSetRole** service roles.

     For information about how AWS Control Tower uses these roles to perform landing zone API calls, see [Step 1: Configure your landing zone](#).

```
Parameters:
  LoggingAccountEmail:
    Type: String
    Description: The email Id for centralized logging account
  LoggingAccountName:
    Type: String
    Description: Name for centralized logging account
  SecurityAccountEmail:
    Type: String
    Description: The email Id for security roles account
  SecurityAccountName:
    Type: String
    Description: Name for security roles account
Resources:
  MyOrganization:
    Type: 'AWS::Organizations::Organization'
    Properties:
      FeatureSet: ALL
  LoggingAccount:
    Type: 'AWS::Organizations::Account'
    Properties:
      AccountName: !Ref LoggingAccountName
      Email: !Ref LoggingAccountEmail
```

```
  SecurityAccount:
    Type: 'AWS::Organizations::Account'
    Properties:
      AccountName: !Ref SecurityAccountName
      Email: !Ref SecurityAccountEmail
  AWSControlTowerAdmin:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: AWSControlTowerAdmin
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: controltower.amazonaws.com
            Action: 'sts:AssumeRole'
      Path: '/service-role/'
      ManagedPolicyArns:
        - !Sub >-
          arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSControlTowerServiceRolePolicy
  AWSControlTowerAdminPolicy:
    Type: 'AWS::IAM::Policy'
    Properties:
      PolicyName: AWSControlTowerAdminPolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action: 'ec2:DescribeAvailabilityZones'
            Resource: '*'
      Roles:
        - !Ref AWSControlTowerAdmin
  AWSControlTowerCloudTrailRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: AWSControlTowerCloudTrailRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: cloudtrail.amazonaws.com
            Action: 'sts:AssumeRole'
```

```
      Path: '/service-role/'
  AWSControlTowerCloudTrailRolePolicy:
    Type: 'AWS::IAM::Policy'
    Properties:
      PolicyName: AWSControlTowerCloudTrailRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Action:
              - 'logs:CreateLogStream'
              - 'logs:PutLogEvents'
            Resource: !Sub >-
              arn:${AWS::Partition}:logs:*:*:log-group:aws-controltower/
CloudTrailLogs:*
            Effect: Allow
      Roles:
        - !Ref AWSControlTowerCloudTrailRole
  AWSControlTowerConfigAggregatorRoleForOrganizations:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: AWSControlTowerConfigAggregatorRoleForOrganizations
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: config.amazonaws.com
            Action: 'sts:AssumeRole'
      Path: '/service-role/'
      ManagedPolicyArns:
        - !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSConfigRoleForOrganizations
  AWSControlTowerStackSetRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: AWSControlTowerStackSetRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: cloudformation.amazonaws.com
            Action: 'sts:AssumeRole'
      Path: '/service-role/'
```

```
  AWSControlTowerStackSetRolePolicy:
    Type: 'AWS::IAM::Policy'
    Properties:
      PolicyName: AWSControlTowerStackSetRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Action: 'sts:AssumeRole'
            Resource: !Sub 'arn:${AWS::Partition}:iam::*:role/
AWSControlTowerExecution'
            Effect: Allow
      Roles:
        - !Ref AWSControlTowerStackSetRole

Outputs:
  LogAccountId:
    Value:
      Fn::GetAtt: LoggingAccount.AccountId
    Export:
      Name: LogAccountId
  SecurityAccountId:
    Value:
      Fn::GetAtt: SecurityAccount.AccountId
    Export:
      Name: SecurityAccountId
```

Create a new landing zone using AWS CloudFormation

From the AWS CloudFormation console or using the AWS CLI, deploy the following AWS CloudFormation template to create a landing zone.

```
Parameters:
  Version:
    Type: String
    Description: The version number of Landing Zone
  GovernedRegions:
    Type: Array
    Description: List of governed regions
  SecurityOuName:
    Type: String
    Description: The security Organizational Unit name
  SandboxOuName:
```

```
    Type: String
    Description: The sandbox Organizational Unit name
  CentralizedLoggingAccountId:
    Type: String
    Description: The AWS account ID for centralized logging
  SecurityAccountId:
    Type: String
    Description: The AWS account ID for security roles
  LoggingBucketRetentionPeriod:
    Type: Number
    Description: Retention period for centralized logging bucket
  AccessLoggingBucketRetentionPeriod:
    Type: Number
    Description: Retention period for access logging bucket
  KMSKey:
    Type: String
    Description: KMS key ARN used by CloudTrail and Config service to encrypt data in
 logging bucket
Resources:
  MyLandingZone:
    Type: 'AWS::ControlTower::LandingZone'
    Properties:
      Version:
        Ref: Version
      Tags:
        - Key: "keyname1"
          Value: "value1"
        - Key: "keyname2"
          Value: "value2"
      Manifest:
        governedRegions:
          Ref: GovernedRegions
        organizationStructure:
          security:
            name:
              Ref: SecurityOuName
          sandbox:
            name:
              Ref: SandboxOuName
        centralizedLogging:
          accountId:
            Ref: CentralizedLoggingAccountId
          configurations:
            loggingBucket:
```

```
              retentionDays:
                Ref: LoggingBucketRetentionPeriod
            accessLoggingBucket:
              retentionDays:
                Ref: AccessLoggingBucketRetentionPeriod
            kmsKeyArn:
              Ref: KMSKey
          enabled: true
        securityRoles:
          accountId:
            Ref: SecurityAccountId
        accessManagement:
          enabled: true
```

Manage an existing landing zone using AWS CloudFormation

You can use AWS CloudFormation to manage a landing zone that you have already launched by importing the landing zone in a new or existing AWS CloudFormation stack. Review Bringing existing resources into CloudFormation management for details and instructions.

To detect and resolve drift within a landing zone, you can use the AWS Control Tower console, the AWS CLI, or the `ResetLandingZone` API.

Next steps

Now that your landing zone is set up, it's ready for use.

To learn more about how you can use AWS Control Tower, see the following topics:

- For recommended administrative practices, see Best Practices.
- You can set up IAM Identity Center users and groups with specific roles and permissions. For recommendations, see Recommendations for setting up groups, roles, and policies .
- To begin enrolling organizations and accounts from your AWS Organizations deployments, see Govern existing organizations and accounts.
- Your end users can provision their own AWS accounts in your landing zone using Account Factory. For more information, see Permissions for configuring and provisioning accounts.
- To assure Compliance Validation for AWS Control Tower, your central cloud administrators can review log archives in the Log Archive account, and designated third-party auditors can review audit information in the Audit (shared) account, which is a member of the Security OU.

- To learn more about the capabilities of AWS Control Tower, see Related information.
- From time to time, you may need to update your landing zone to get the latest backend updates, the latest controls, and to keep your landing zone up-to-date. For more information, see Configuration update management in AWS Control Tower.
- If you encounter issues while using AWS Control Tower, see Troubleshooting.

> **Important**
>
> If you have not yet enabled MFA for your account's root user, do so now. For more information about best practices for the root user, see Best practices to protect your account's root user.

Limitations and quotas in AWS Control Tower

This chapter covers the AWS service limitations and quotas that you should keep in mind as you use AWS Control Tower. If you're unable to set up your landing zone due to a service quota issue, contact AWS Support.

For more information about limitations that are specific to controls, see Control limitations.

> **The Controls Reference Guide**
>
> Detailed information about AWS Control Tower controls has been moved to the AWS Control Tower Controls Reference Guide.

Known limitations in AWS Control Tower

This section describes known limitations and unsupported use cases in AWS Control Tower.

- AWS Control Tower has overall *concurrency limitations*. In general, one operation at a time is permitted. Two exceptions to this limitation are allowed:
  - Optional controls can be activated and deactivated concurrently, through an asynchronous process. Up to one hundred (100) control-related operations at a time can be in progress, in total, no matter if they are called from the console or from an API.
  - Accounts can be provisioned, updated, and enrolled concurrently in Account Factory, through an asynchronous process, with up to five (5) account-related operations in progress simultaneously. Unmanaging accounts must be performed one account at a time.
- Email addresses of shared accounts in the Security OU can be changed, but you must update your landing zone to see these changes in the AWS Control Tower console.
- A limit of five (5) SCPs per OU applies to OUs in your AWS Control Tower landing zone.
- AWS Control Tower supports up to 10,000 accounts in your landing zone's organization, divided among all of your OUs.
- Existing OUs with over 1000 directly nested accounts cannot be registered or re-registered in AWS Control Tower. For more information about limitations with registering OUs, see Limitations based on underlying AWS services.
- **Customizations for AWS Control Tower (CfCT)** is unavailable in these AWS Regions, because some dependencies are not available:

  - Europe (Zurich), eu-central-2
  - Europe (Spain), eu-south-2
  - Canada West (Calgary) ca-west-1
  - Asia Pacific (Melbourne), ap-southeast-4
  - Asia Pacific (Malaysia), ap-southeast-5
  - Asia Pacific (Thailand), ap-southeast-7
  - Mexico (Central), mx-central-1

  You can deploy and manage resources in these Regions with CfCT, if you deploy CfCT to your AWS Control Tower home Region, but you cannot build CfCT in these Regions.

- **AWS Control Tower Account Factory for Terraform (AFT)** is not available in the following AWS Regions, because some dependencies are not available:
  - Europe (Zurich), eu-central-2
  - Europe (Spain), eu-south-2
  - Canada West (Calgary), ca-west-1
  - Asia Pacific (Melbourne), ap-southeast-4
  - Asia Pacific (Malaysia), ap-southeast-5
  - Asia Pacific (Thailand), ap-southeast-7
  - Mexico (Central), mx-central-1
- **AWS Control Tower Account Factory for Terraform (AFT)** cannot be deployed by new AFT customers in the following Regions, because AWS CodeConnections is not available to connect to a third-party version control system (VCS):
  - Asia Pacific (Hong Kong), Africa (Cape Town), Middle East (Bahrain), Europe (Zurich), Asia Pacific (Jakarta), Asia Pacific (Hyderabad), Asia Pacific (Osaka), Asia Pacific (Melbourne), Israel (Tel Aviv), Europe (Spain), Middle East (UAE), Asia Pacific (Thailand), Mexico (Central)
- The following Regions do not support **AWS Service Catalog**.
  - Canada West (Calgary), ca-west-1
  - Asia Pacific (Malaysia), ap-southeast-5
  - Asia Pacific (Thailand), ap-southeast-7
  - Mexico (Central), mx-central-1

> **Note**
>
> Regions that do not support Service Catalog do not support Account Factory Customizations for AWS Control Tower (AFC).

For more information about AWS Control Tower functionality in Regions that do not support AWS Service Catalog, see AWS Control Tower available in AWS Canada West (Calgary).

- When calling a control API to activate or deactivate a control, the limit for `EnableControl` and `DisableControl` updates in AWS Control Tower is one hundred (100) concurrent operations. Ten operations (10) can be in progress simultaneously, with the remaining operations queued. You may need to adjust your code to wait for completions.
- When you provision accounts through **Account Factory Customizations (AFC)**, with blueprints that are based in Terraform, you can deploy those blueprints to only one AWS Region. By default, AWS Control Tower deploys to the home Region.

Request a quota increase

The Service Quotas console provides information about AWS Control Tower quotas. You can use the Service Quotas console to view the default service quotas or to request quota increases for adjustable quotas.

> **Note**
>
> Newly created accounts and organizations might experience a quota below the default of 10 accounts.

The following quotas can be viewed through the Service Quotas console

- *Concurrent account operations quota* : The maximum number of concurrent account operations that can be performed at the same time. Default: 5, Maximum: 10, adjustable
- *Number of accounts in a single OU* : The maximum number of AWS Control Tower managed accounts that can be present in one OU. If you add accounts beyond this limit, the OU registration process in AWS Control Tower cannot be performed. To learn more about the

number of accounts per OU, review Limitations based on underlying AWS services in the AWS Control Tower documentation. Default: 1000, not adjustable.

- *Concurrent operations for organizational units (OUs)* : The maximum number of concurrent OU-related operations that can be performed at the same time. Default: 1, not adjustable.

For example, you can request a quota increase from five of up to ten concurrent account-related operations. Some AWS Control Tower performance characteristics may change after a quota increase. For example, it may take longer to update an OU when you have more accounts in it. Or, it may take longer to complete an action on an OU with five SCPs than with three SCPs.

> **Note**
>
> A service quota increase request may require up to two days before it takes effect. Be sure to request the quota increase from your AWS Control Tower home Region.

As an alternative, you can contact AWS Support to request a quota increase for some resources in AWS Control Tower. Or you can view the video that follows, and learn how to automate certain service quota increases.

Video: Automate requests for service quota increases, in services related to AWS Control Tower

This video (7:24) describes how to automate service quota increases for related, integrated AWS services, based on deployments in AWS Control Tower. It also shows how to automate enrollment of new accounts into AWS Enterprise support for your organization. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Quota Increases in AWS Control Tower.

When provisioning new accounts in this environment, you can use lifecycle events to trigger automated requests for service quota increases in specified AWS Regions.

More information about AWS quotas is available in the AWS General Reference.

Control limitations

AWS Control Tower assists you with maintaining a secure, multi-account environment on AWS by means of controls, which are implemented in various forms, such as service control policies (SCPs), AWS Config rules, and AWS CloudFormation hooks.

The Controls Reference Guide

Detailed information about AWS Control Tower controls has been moved to the AWS Control Tower Controls Reference Guide.

If you modify AWS Control Tower resources, such as an SCP, or remove any AWS Config resource, such as a Config recorder or aggregator, AWS Control Tower can no longer guarantee that the controls are functioning as designed. Therefore, the security of your multi-account environment may be compromised. The AWS shared responsibility model of security is applicable to any such changes you may make.

Note

AWS Control Tower helps maintain the integrity of your environment by resetting the SCPs of the preventive controls to their standard configuration when you update your landing zone. Changes that you may have made to SCPs are replaced by the standard version of the control, by design.

Limitations by Region

Some controls in AWS Control Tower do not operate in certain AWS Regions where AWS Control Tower is available, because those Regions do not support the required underlying functionality. As a result, when you deploy that control, it may not be operating in all Regions that you govern with AWS Control Tower. This limitation affects certain detective controls, certain proactive controls, and certain controls in the **Security Hub Service-managed Standard: AWS Control Tower**. For more information about Regional availability, see the Security Hub controls. Also see the Regional services list documentation and the Security Hub controls reference documentation.

Control behavior also is limited in case of *mixed governance*. For more information, see Avoid mixed governance when configuring Regions.

For more information about how AWS Control Tower manages the limitations of Regions and controls, see Considerations for activating AWS opt-in Regions.

> **Note**
>
> For the most updated information about controls and Region support, we recommend that you call the `GetControl` and `ListControls` API operations.

Find available controls and Regions

You can view the available Regions for each control in the AWS Control Tower console. You can view the available Regions programmatically with the `GetControl` and `ListControls` APIs from AWS Control Catalog.

Also see the reference table of AWS Control Tower controls and supported Regions, Control availability by Region, in the *AWS Control Tower Controls Reference Guide*.

For information about AWS Security Hub controls from the **Service-Managed Standard: AWS Control Tower** that are not supported in certain AWS Regions, see "Unsupported Regions" in the Security Hub standard.

The following table shows specific proactive controls that are not supported in certain AWS Regions.

Control identifier	Non-deployable regions
`CT.DAX.PR.2`	ap-southeast-5, ca-west-1, us-west-1
`CT.REDSHIFT.PR.5`	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1

The following table shows AWS Control Tower detective controls that are not supported in certain AWS Regions.

Control identifier	Non-deployable regions
`API_GW_CACHE_ENABLED_AND_ENCRYPTED`	ap-southeast-5, ca-west-1, mx-central-1

Control identifier	Non-deployable regions
APPSYNC_ASSOCIATED_WITH_WAF	af-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AURORA_LAST_BACKUP_RECOVERY_POINT_CREATED	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AURORA_RESOURCES_PROTECTED_BY_BACKUP_PLAN	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AUTOSCALING_CAPACITY_REBALANCING	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED	ap-northeast-3, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, il-central-1
AWS-GR_DMS_REPLICATION_NOT_PUBLIC	af-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1
AWS-GR_EBS_OPTIMIZED_INSTANCE	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK	eu-south-2
AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP	ap-northeast-3
AWS-GR_EC2_VOLUME_INUSE_CHECK	ap-southeast-5, ca-west-1

Control identifier	Non-deployable regions
AWS-GR_EKS_ENDPOINT_NO_PUBL IC_ACCESS	ap-southeast-5, ca-west-1
AWS-GR_ELASTICSEARCH_IN_VPC_ONLY	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-2, il-central-1
AWS-GR_EMR_MASTER_NO_PUBLIC_IP	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1
AWS-GR_ENCRYPTED_VOLUMES	af-south-1, ap-northeast-3, eu-south-1, il-central-1
AWS-GR_IAM_USER_MFA_ENABLED	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AWS-GR_LAMBDA_FUNCTION_PUBL IC_ACCESS_PROHIBITED	eu-south-2
AWS-GR_MFA_ENABLED_FOR_IAM_ CONSOLE_ACCESS	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
AWS-GR_NO_UNRESTRICTED_ROUT E_TO_IGW	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2
AWS-GR_RDS_INSTANCE_PUBLIC_ ACCESS_CHECK	ap-south-2, eu-south-2
AWS-GR_RDS_SNAPSHOTS_PUBLIC _PROHIBITED	af-south-1, ap-southeast-4, eu-central-2, eu-south-1, eu-south-2, il-central-1

Control identifier	Non-deployable regions
AWS-GR_RDS_STORAGE_ENCRYPTED	eu-central-2, eu-south-2
AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK	ap-south-2, ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2, mx-central-1
AWS-GR_RESTRICTED_SSH	af-south-1, eu-south-1
AWS-GR_ROOT_ACCOUNT_MFA_ENABLED	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, me-central-1, mx-central-1
AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC	eu-central-2, eu-south-2, il-central-1
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
AWS-GR_SSM_DOCUMENT_NOT_PUBLIC	ap-southeast-5, ca-west-1, il-central-1
AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED	ap-northeast-3
BACKUP_PLAN_MIN_FREQUENCY_AND_MIN_RETENTION_CHECK	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
BACKUP_RECOVERY_POINT_MANUAL_DELETION_DISABLED	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
BACKUP_RECOVERY_POINT_MINIMUM_RETENTION_CHECK	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ACCOUNT.DT.1	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2
CONFIG.APPCONFIG.DT.1	ap-south-2, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPCONFIG.DT.2	ap-south-2, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPCONFIG.DT.3	ap-south-2, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPCONFIG.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.APPMESH.DT.1	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPMESH.DT.2	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPMESH.DT.3	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.APPMESH.DT.4	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPMESH.DT.5	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPMESH.DT.6	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPMESH.DT.7	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.APPSYNC.DT.2	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, mx-central-1, us-east-2, us-west-1, us-west-2
CONFIG.APPSYNC.DT.3	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, mx-central-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.APPSYNC.DT.4	af-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.APPSYNC.DT.5	ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.ATHENA.DT.1	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ATHENA.DT.2	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, mx-central-1
CONFIG.AUTOSCALING.DT.1	ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.BACKUP-GATEWAY.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.BACKUP-GATEWAY.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.BACKUP-GATEWAY.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.CLOUDFORMATION.DT.1	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.CLOUDFRONT.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.10	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.CLOUDFRONT.DT.11	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.2	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.3	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.CLOUDFRONT.DT.4	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.5	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.6	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.CLOUDFRONT.DT.7	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.8	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDFRONT.DT.9	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.CLOUDTRAIL.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.CLOUDTRAIL.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.CLOUDTRAIL.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.CLOUDTRAIL.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.CLOUDTRAIL.DT.5	ap-southeast-5, ca-west-1
CONFIG.CLOUDWATCH.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.CODEBUILD.DT.2	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.CODEBUILD.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.CODEDEPLOY.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.CODEDEPLOY.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.CODEDEPLOY.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.CODEPIPELINE.DT.1	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.CODEPIPELINE.DT.2	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.COGNITO-IDP.DT.1	af-south-1, ap-east-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.DATASYNC.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.DAX.DT.1	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.DMS.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DMS.DT.2	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DMS.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DMS.DT.4	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DMS.DT.5	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DMS.DT.6	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DYNAMODB.DT.1	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-1, eu-south-2

Control identifier	Non-deployable regions
CONFIG.DYNAMODB.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DYNAMODB.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.DYNAMODB.DT.4	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2
CONFIG.DYNAMODB.DT.5	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, eu-north-1, eu-south-1, eu-south-2, me-south-1
CONFIG.DYNAMODB.DT.6	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.DYNAMODB.DT.7	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1
CONFIG.EC2.DT.10	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.11	ap-southeast-5, ca-west-1
CONFIG.EC2.DT.13	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.EC2.DT.14	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.15	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.16	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.17	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2
CONFIG.EC2.DT.19	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2
CONFIG.EC2.DT.2	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, us-east-2
CONFIG.EC2.DT.20	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.EC2.DT.21	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.22	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.EC2.DT.5	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EC2.DT.6	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1
CONFIG.EC2.DT.7	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.EC2.DT.8	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.EC2.DT.9	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ECS.DT.1	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ECS.DT.2	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ECS.DT.3	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ECS.DT.4	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.EKS.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.EKS.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.EKS.DT.4	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-centra l-1

Control identifier	Non-deployable regions
CONFIG.ELASTICACHE.DT.1	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICACHE.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICACHE.DT.3	ap-southeast-4, ap-southeast-5, ap-southe ast-7, ca-west-1, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICBEANSTALK.DT.1	ap-northeast-3, ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.ELASTICFILESYSTEM.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.ELASTICFILESYSTEM.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.ELASTICFILESYSTEM.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICFILESYSTEM.DT.4	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.ELASTICFILESYSTEM.DT.5	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICFILESYSTEM.DT.6	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.ELASTICLOADBALANCING.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.ELASTICLOADBALANCING.DT.3	ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.EMR.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.EMR.DT.2	ap-southeast-5, ca-west-1
CONFIG.EVENTS.DT.1	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.EVENTS.DT.2	af-south-1, ap-east-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.EVIDENTLY.DT.1	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-south-1, eu-south-2, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.EVIDENTLY.DT.2	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-south-1, eu-south-2, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.EVIDENTLY.DT.3	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-south-1, eu-south-2, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.FIREHOSE.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.FRAUDDETECTOR.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1

Control identifier	Non-deployable regions
CONFIG.FRAUDDETECTOR.DT.2	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.FRAUDDETECTOR.DT.3	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.FRAUDDETECTOR.DT.4	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.FSX.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.FSX.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.FSX.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.FSX.DT.4	ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.FSX.DT.5	ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GLUE.DT.1	ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GLUE.DT.2	ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GUARDDUTY.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.GUARDDUTY.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GUARDDUTY.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GUARDDUTY.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GUARDDUTY.DT.5	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.GUARDDUTY.DT.6	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.GUARDDUTY.DT.7	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.IAM.DT.1	ap-southeast-5, ca-west-1, il-central-1
CONFIG.IAM.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.IAM.DT.3	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.IAM.DT.4	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.IAM.DT.5	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.IAM.DT.6	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1

Control identifier	Non-deployable regions
`CONFIG.INSPECTOR.DT.1`	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2, il-central-1, me-central-1, mx-central-1
`CONFIG.INSPECTOR.DT.2`	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-south-1, eu-south-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
`CONFIG.INSPECTOR.DT.3`	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2, il-central-1, me-central-1, mx-central-1
`CONFIG.INSPECTOR.DT.4`	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2, il-central-1, me-central-1, mx-central-1
`CONFIG.IOTSITEWISE.DT.1`	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
`CONFIG.IOTSITEWISE.DT.2`	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1

Control identifier	Non-deployable regions
CONFIG.IOTSITEWISE.DT.3	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.IOTSITEWISE.DT.4	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.IOTSITEWISE.DT.5	af-south-1, ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-west-1
CONFIG.KINESIS.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.KMS.DT.1	ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-1, eu-south-2
CONFIG.KMS.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.LAMBDA.DT.1	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2

Control identifier	Non-deployable regions
CONFIG.LAMBDA.DT.2	ap-southeast-5, ca-west-1
CONFIG.LOGS.DT.1	ap-southeast-5, ca-west-1
CONFIG.LOGS.DT.2	ap-northeast-3, eu-south-2
CONFIG.MACIE.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, me-central-1, mx-central-1
CONFIG.MACIE.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.MQ.DT.1	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, mx-central-1
CONFIG.MQ.DT.3	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, mx-central-1
CONFIG.MQ.DT.5	af-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.MSK.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.NETWORK-FIREWALL.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.NETWORK-FIREWALL.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1

Control identifier	Non-deployable regions
CONFIG.NETWORK-FIREWALL.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.OPENSEARCH.DT.1	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.OPENSEARCH.DT.2	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.RDS.DT.1	af-south-1, ap-east-1, eu-north-1, eu-south-1, eu-south-2, il-central-1, me-south-1, sa-east-1
CONFIG.RDS.DT.10	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.12	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.13	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-2, il-central-1, mx-central-1
CONFIG.RDS.DT.14	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.15	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1

Control identifier	Non-deployable regions
CONFIG.RDS.DT.16	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.RDS.DT.17	eu-south-2, il-central-1, me-south-1, sa-east-1
CONFIG.RDS.DT.18	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.RDS.DT.20	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.RDS.DT.3	af-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, us-east-2
CONFIG.RDS.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.6	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1, us-west-1

Control identifier	Non-deployable regions
CONFIG.RDS.DT.8	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.RDS.DT.9	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-1, eu-south-2, mx-central-1
CONFIG.REDSHIFT.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.REDSHIFT.DT.2	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.REDSHIFT.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.REDSHIFT.DT.4	af-south-1, ap-southeast-5, ap-southeast-7, ca-west-1, eu-south-1, il-central-1, mx-central-1
CONFIG.ROUTE53.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1 , eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.S3.DT.10	ap-northeast-3, ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, me-south-1, mx-central-1
CONFIG.S3.DT.11	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.S3.DT.12	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.S3.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.S3.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.S3.DT.4	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.S3.DT.5	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.S3.DT.6	ap-south-2, ap-southeast-5, ca-west-1, eu-south-2
CONFIG.S3.DT.7	af-south-1, ap-east-1, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.S3.DT.9	ap-northeast-3, eu-south-2

Control identifier	Non-deployable regions
CONFIG.SAGEMAKER.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.SAGEMAKER.DT.2	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.SAGEMAKER.DT.3	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.SECRETSMANAGER.DT.1	ap-southeast-5, ca-west-1
CONFIG.SECURITYHUB.DT.1	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1
CONFIG.SERVICECATALOG.DT.1	ap-southeast-3, ap-southeast-5, ap-southeast-7, ca-west-1, il-central-1, mx-central-1
CONFIG.SES.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1

Control identifier	Non-deployable regions
CONFIG.SHIELD.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.SHIELD.DT.2	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.SNS.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.SNS.DT.2	eu-central-2, eu-south-2, il-central-1, me-central-1
CONFIG.SNS.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.SSM.DT.1	ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-5, ca-west-1, eu-south-2

Control identifier	Non-deployable regions
CONFIG.STEPFUNCTIONS.DT.1	ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.STORAGEGATEWAY.DT.1	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.STORAGEGATEWAY.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.STORAGEGATEWAY.DT.3	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.TRANSFER.DT.1	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.WAF.DT.1	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.WAF.DT.2	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1 , eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.WAF.DT.3	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1 , eu-central-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2 , eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2
CONFIG.WAF.DT.4	af-south-1, ap-east-1, ap-northeast-1, ap-northeast-2, ap-northeast-3, ap-south-1, ap-south-2, ap-southeast-1, ap-southeast-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-central-1, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-1, eu-west-2, eu-west-3, il-central-1, me-central-1, me-south-1, mx-central-1, sa-east-1, us-east-2, us-west-1, us-west-2

Control identifier	Non-deployable regions
CONFIG.WAFV2.DT.1	af-south-1, ap-northeast-3, ap-south-2, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-south-1, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.WAFV2.DT.2	ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1 , eu-central-2, eu-south-2, il-central-1, me-central-1, mx-central-1
CONFIG.WAFV2.DT.3	ap-southeast-5, ap-southeast-7, ca-west-1, mx-central-1
CONFIG.WORKSPACES.DT.1	ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, us-east-2, us-west-1
CONFIG.WORKSPACES.DT.2	ap-east-1, ap-northeast-3, ap-south-2, ap-southeast-3, ap-southeast-4, ap-southeast-5, ap-southeast-7, ca-west-1, eu-central-2, eu-north-1, eu-south-1, eu-south-2, eu-west-3 , il-central-1, me-central-1, me-south-1, mx-central-1, us-east-2, us-west-1

Limitations based on underlying AWS services

This page describes limitations that you may encounter due to limitations in other AWS services, and how AWS Control Tower works with those services.

General guidelines

As a general rule, we expect that the number of accounts supported when registering an OU diminishes as you increase the number of Regions governed, and the number of controls enabled, for that OU. These general guidelines assume that you have 15 optional controls enabled. If you have more or fewer controls enabled on your OU, the limits on accounts per OU will differ when registering.

- With 15 governed Regions, OUs of up to 1000 accounts are supported.
- With 16 to 21 governed Regions, the maximum supported OU size is in the range of 600-1000 accounts.
- With 22 governed Regions, OUs of up to 680 accounts are supported.
- With 23 or more governed Regions, the maximum supported OU size is less than 680 accounts.

In case of error

If registration fails, you can try to **Re-register** the OU. Also, you can make the OU smaller by using a nested OU or by moving acounts to another OU.

> **Note**
>
> The mandatory controls that AWS Control Tower always enforces are not counted toward the number of controls you've enabled on an OU, for purposes of registration.

AWS CloudFormation stack set limitations

If you plan to register a large number of accounts across multiple AWS Regions, you may encounter limits created by AWS CloudFormation stack sets on the overall size of an organization. You can estimate the limitation with this formula:

Number of managed accounts in the organization x Number of governed Regions <= 150,000

This limitation becomes apparent during the OU registration process. For example, if 15 Regions are governed, and 15 optional controls are enabled, the limit for registering the OU is 1000 accounts. However, if you require to register OUs with more than 1000 accounts, or if you have a large number of optional controls enabled, you must reduce the number of governed Regions below 15. This reduction is due to stack set limitations.

AWS Config limitations

If you plan to register OUs with a large number of accounts, you may encounter limits with [the maximum number of accounts that AWS Config allows to be created or deleted each week](), across all aggregators. Enrolled accounts do not count toward this limit: You may enroll up to 1000 new accounts into AWS Control Tower each week.

First-time limitations for accounts and opt-in Regions

If you plan to register OUs with a large number of accounts across multiple opt-in Regions *for the first time*, you may encounter limitations due to Account Management quotas, which can lead to prolonged latency. Errors may occur during OU registration due to latency.

Regional differences for AWS Control Tower functionality

Certain differences exist in the behavior of AWS Control Tower across AWS Regions, because AWS Control Tower orchestrates the behavior of other AWS services. For example:

- AWS Service Catalog is not available in all AWS Regions where AWS Control Tower is available, which changes the behavior of Account Factory in those Regions.
- In certain Regions, Account Factory Customizations (AFC) is not available because Service Catalog is not available to support the underlying functionality for blueprints.
- Certain controls are not available in all AWS Regions due to lack of underlying functionality.
- AFT and CfCT are not available in all AWS Regions due to lack of underlying functionality.

To make the best determination of behavior for your AWS Control Tower environment, ascertain your home Region. Then, evaluate the following items. For more details, see Limitations and quotas in AWS Control Tower.

- Is AWS Service Catalog available in your desired home Region?
- Are the controls available that you require? See Control limitations.
- Is IAM Identity Center available in your desired home Region?

Deployable Regions for controls

AWS Control Tower cannot activate certain controls when you deploy them in certain Regions, due to lack of underlying dependencies. You can find the most updated information about the deployable Regions for any control by calling the `ListControls` and `GetControl` APIs. You also can view the deployable Regions in the AWS Control Tower console.

When you activate a control on an OU that's governed by AWS Control Tower, the control's effective area is the *intersection* of your AWS Control Tower governed Regions with the control's deployable Regions.

For example, a control can be enabled on an OU that operates in governed Regions X, Y and Z. But after it is enabled, the same control is deployed only on Regions X and Z, because the control itself does not support Region Y.

It's important to monitor the relationships among controls that you deploy and Regions where you operate workloads in AWS Control Tower, so that you don't experience gaps in protection of your AWS resources.

How to check your protected Regions

- In the AWS Control Tower console, you can view the enabled controls and Regions in the **Enabled controls** section.
- If you call the `GetEnabledControl` API, the **targetRegions** parameter will show only those Regions where you can deploy the control effectively, not the non-deployable Regions.

The AWS Control Tower Controls Reference Guide

Detailed information about controls in AWS Control Tower has moved to the *AWS Control Tower Controls Reference Guide*.

Best practices for AWS Control Tower administrators

This topic is intended primarily for management account administrators.

Management account administrators are responsible for explaining some tasks that AWS Control Tower controls prevent their member account administrators from doing. This topic describes some best practices and procedures for transferring this knowledge, and it gives other tips for setting up and maintaining your AWS Control Tower environment efficiently.

Explaining access to users

The AWS Control Tower console is available only to users with the management account administrator permissions. Only these users can perform administrative work within your landing zone. In accordance with best practices, this means that the majority of your users and member account administrators will never see the AWS Control Tower console. As a member of the management account administrator group, it's your responsibility to explain the following information to the users and administrators of your member accounts, as appropriate.

- Explain which AWS resources that users and administrators have access to within the landing zone.
- List the preventive controls that apply to each organizational unit (OU) so that the other administrators can plan and execute their AWS workloads accordingly.

Explaining resource access

Some administrators and other users may need an explanation of the AWS resources to which they have access to within your landing zone. This access can include programmatic access and console-based access. Generally speaking, read access and write access for AWS resources is allowed. To perform work within AWS, your users require some level of access to the specific services they need to do their jobs.

Some users, such as your AWS developers, may need to know about the resources to which they have access, so they can create engineering solutions. Other users, such as the end users of the applications that run on AWS services, do not need to know about AWS resources within your landing zone.

AWS offers tools to identify the scope of a user's AWS resource access. After you identify the scope of a user's access, you can share that information with the user, in accordance with your organization's information management policies. For more information about these tools, see the links that follow.

- **AWS access advisor** – The AWS Identity and Access Management (IAM) access advisor tool lets you determine the permissions that your developers have by analyzing the last timestamp when an IAM entity, such as a user, role, or group, called an AWS service. You can audit service access and remove unnecessary permissions, and you can automate the process if needed. For more information, see our AWS Security blog post.
- **IAM policy simulator** – With the IAM policy simulator, you can test and troubleshoot IAM-based and resource-based policies. For more information, see Testing IAM Policies with the IAM Policy Simulator.
- **AWS CloudTrail logs** – You can review AWS CloudTrail logs to see actions taken by a user, role, or AWS service. For more information about CloudTrail, see the AWS CloudTrail User Guide.

  Actions taken by AWS Control Tower landing zone administrators are viewable in the landing zone management account. Actions taken by member account administrators and users are viewable in the shared log archive account.

  You can view a summary table of AWS Control Tower events in the Activities page.

Explaining preventive controls

A preventive control ensures that your organization's accounts maintain compliance with your corporate policies. The status of a preventive control is either **enforced** or **not-enabled**. A preventive control prevents policy violations by using service control policies (SCPs). In comparison, a detective control informs you of various events or states that exist, by means of defined AWS Config rules.

Some of your users, such as AWS developers, may need to know about the preventive controls that apply to any accounts and OUs they use, so they can create engineering solutions. The following procedure offers some guidance on how to provide this information for the right users, according to your organization's information management policies.

> **Note**
>
> This procedure assumes you've already created at least one child OU within your landing zone, as well as at least one AWS IAM Identity Center user.

To show preventive controls for users with a need to know

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower/.
2. From the left navigation, choose **Organization**.
3. From the table, choose the **name** of one of the OUs for which your user needs information about the applicable controls.
4. Note the name of the OU and the controls that apply to this OU.
5. Repeat the previous two steps for each OU about which your user needs information.

For detailed information about the controls and their functions, see About controls in AWS Control Tower.

Plan your AWS Control Tower landing zone

When you go through the setup process, AWS Control Tower launches a key resource associated with your account, called a *landing zone*, which serves as a home for your organizations and their accounts.

> **Note**
>
> You can have one landing zone per organization.

For information about some best practices to follow when you plan and set up your landing zone, see AWS multi-account strategy for your AWS Control Tower landing zone.

Ways to Set Up AWS Control Tower

You can set up an AWS Control Tower landing zone in an existing organization, or you can start by creating a new organization that contains your AWS Control Tower landing zone.

- Launch AWS Control Tower in an Existing Organization: This section is for customers who have existing AWS Organizations ready to bring into governance by AWS Control Tower.
- Launch AWS Control Tower in a New Organization: This section is for customers without existing AWS Organizations, OUs, and accounts.

> **Note**
>
> If you already have an AWS Organizations landing zone, you can extend AWS Control Tower governance from the existing landing zone to some or all of your existing OUs and accounts within an organization. See Govern existing organizations and accounts.

Compare functionality

Here's a brief comparison of the differences between adding AWS Control Tower to an existing organization or extending AWS Control Tower governance to OUs and accounts. Also, some special considerations apply if you are moving to AWS Control Tower from the AWS Landing Zone solution.

About Adding to an Existing Organization: Adding AWS Control Tower to an existing organization is something you can accomplish within the AWS console. In this case, you've already got an organization that you've created in the AWS Organizations service, that organization is not currently registered with AWS Control Tower, and you want to *add a landing zone afterward*.

When you *add* a landing zone to an existing organization, AWS Control Tower sets up a parallel structure, at the AWS Organizations level. It doesn't change the OUs and accounts within your existing organization.

About Extending Governance: Extending governance applies to specific OUs and accounts *within a single organization that's already registered* with AWS Control Tower, which means that a landing zone already exists for that organization. Extending governance means that AWS Control Tower controls are extended so that their constraints apply to the specific OUs and accounts within that registered organization. In this case, you're not launching a new landing zone, you're only expanding the current landing zone for your organization.

> **Important**
>
> Special consideration: If you currently are using the AWS Landing Zone solution (ALZ) for AWS Organizations, check with your AWS solutions architect before you try to enable

AWS Control Tower in your organization. AWS Control Tower cannot perform pre-checks that determine whether AWS Control Tower may interfere with your current landing zone deployment. For more information, see Walkthrough: Move from ALZ to AWS Control Tower. Also, for information about moving accounts from one landing zone to another, see If the account does not meet the prerequisites

Launch AWS Control Tower in an Existing Organization

By setting up an AWS Control Tower landing zone in an existing organization, you can start working immediately, in parallel with your existing AWS Organizations environment. Your other OUs created within AWS Organizations are unchanged, because they are not registered with AWS Control Tower. You can continue to use those OUs and accounts exactly as they are.

AWS Control Tower consolidates by using the management account from your existing organization as its management account. No new management account is needed. You can launch your AWS Control Tower landing zone from your existing management account.

Note

To set up AWS Control Tower on an existing organization, your service limits must allow for the creation of at least two additional accounts.

Effects of adding AWS Control Tower to your existing organization

AWS Control Tower creates two accounts in your organization: an audit account and a logging account. These accounts keep a record of actions taken by your team, in their individual end-user accounts. The **Audit** and **Log archive** accounts appear in the **Security** OU within your AWS Control Tower landing zone.

When you set up your landing zone, the accounts added by AWS Control Tower become part of your existing AWS Organizations, and as such they become part of the billing for your existing organization.

Summary of capabilities

Enabling AWS Control Tower on an existing AWS Organizations organization provides several major enhancements to the organization.

- It allows for unified billing across your organization's groups, because accounts added by AWS Control Tower will become part of your existing organization.
- It gives you the ability to administer all accounts from one management account in your OU.
- It simplifies how you apply and enforce controls that cover security and compliance for existing and new accounts.

> **Important**
>
> Launching your AWS Control Tower landing zone in an existing AWS Organizations organization does not enable you to extend AWS Control Tower governance from that organization to other OUs or accounts that are not registered with AWS Control Tower.

To launch AWS Control Tower in your existing organization, follow the process outlined in Getting started with AWS Control Tower.

For more information about how AWS Control Tower interacts with existing AWS Organizations organizations, see Govern organizations and accounts with AWS Control Tower.

Launch AWS Control Tower in a New Organization

If you're new to AWS Control Tower and you haven't worked with AWS Organizations, the best place to begin is with our Setting up document.

AWS Control Tower sets up an organization for you automatically when you don't have one set up.

AWS multi-account strategy for your AWS Control Tower landing zone

AWS Control Tower customers often seek guidance about how to set up their AWS environment and accounts for best results. AWS has created a unified set of recommendations, called the *multi-account strategy*, to help you make the best use of your AWS resources, including your AWS Control Tower landing zone.

Essentially, AWS Control Tower acts as an orchestration layer that works with other AWS services, which assist you with implementing the AWS multi-account recommendations for AWS accounts and AWS Organizations. After your landing zone is set up, AWS Control Tower continues to assist you with maintaining your corporate policies and security practices across multiple accounts and workloads.

Most landing zones develop over time. As the number of organizational units (OUs) and accounts in your AWS Control Tower landing zone increases, you can extend your AWS Control Tower deployment in ways that help organize your workloads effectively. This chapter provides prescriptive guidance on how to plan and set up your AWS Control Tower landing zone, in alignment with the AWS multi-account strategy, and extend it over time.

For a general discussion about best practices for organizational units, see Best Practices for Organizational Units with AWS Organizations.

AWS multi-account strategy: Best practices guidance

AWS best practices for a well-architected environment recommend that you should separate your resources and workloads into multiple AWS accounts. You can think of AWS accounts as isolated resource containers: they offer workload categorization, as well as blast radius reduction when things go wrong.

Definition of an AWS account

> *An AWS account acts as a resource container and resource isolation boundary.*

> **Note**
>
> An AWS account is not the same as a user account, which is set up through Federation or AWS Identity and Access Management (IAM).

More about AWS accounts

An AWS account provides the ability to isolate resources and to contain security threats for your AWS workloads. An account also provides a mechanism for billing and for governance of a workload environment.

The AWS account is the primary implementation mechanism to provide a resource container for your workloads. If your environment is well-architected, you can manage multiple AWS accounts effectively, and thus, manage multiple workloads and environments.

AWS Control Tower sets up a well-architected environment. It relies upon AWS accounts, along with AWS Organizations, which help govern changes to your environment that can extend across multiple accounts.

Definition of a well-architected environment

> *AWS defines a well-architected environment as one that begins with a landing zone.*

AWS Control Tower offers a landing zone that is set up automatically. It enforces controls to ensure compliance with your corporate guidelines, across multiple accounts in your environment.

Definition of a landing zone

> *The landing zone is a cloud environment that offers a recommended starting point, including default accounts, account structure, network and security layouts, and so forth. From a landing zone, you can deploy workloads that utilize your solutions and applications.*

Guidelines to set up a well-architected environment

The three key components of a well-architected environment, explained in the following sections, are:

- Multiple AWS accounts
- Multiple organizational units (OUs)
- A well-planned structure

Use multiple AWS accounts

One account isn't enough to set up a well-architected environment. By using multiple accounts, you can best support your security goals and business processes. Here are some benefits of using a multi-account approach:

- **Security controls** – Applications have different security profiles, so they require different control policies and mechanisms. For example, it's far easier to talk to an auditor and point to a single account hosting the payment card industry (PCI) workload.
- **Isolation** – An account is a unit of security protection. Potential risks and security threats can be contained within an account without affecting others. Therefore, security needs may require you

to isolate accounts from one another. For example, you may have teams with different security profiles.

- **Many teams –** Teams have different responsibilities and resource needs. By setting up multiple accounts, the teams cannot interfere with one another, as they might when using the same account.
- **Data Isolation –** Isolating data stores to an account helps limit the number of people who have access to data and can manage the data store. This isolation helps prevent unauthorized exposure of highly private data. For example, data isolation helps support compliance with the General Data Protection Regulation (GDPR).
- **Business process –** Business units or products often have completely different purposes and processes. Individual accounts can be established to serve business-specific needs.
- **Billing –** An account is the only way to separate items at a billing level, including things like transfer charges and so forth. The multi-account strategy helps create separate billable items across business units, functional teams, or individual users.
- **Quota allocation –** AWS quotas are set up on a per-account basis. Separating workloads into different accounts gives each account (such as a project) a well-defined, individual quota.

Use multiple organizational units

AWS Control Tower and other account orchestration frameworks can make changes that cross account boundaries. Therefore, the AWS best practices address cross-account changes, which potentially can break an environment or undermine its security. In some cases, changes can affect the overall environment, beyond policies. As a result, we recommend that you should set up at least two mandatory accounts, Production and Staging.

Furthermore, AWS accounts often are grouped into organizational units (OUs), for purposes of governance and control. OUs are designed to handle enforcement of policies across multiple accounts.

Our recommendation is that, at a minimum, you create a pre-production (or Staging) environment that is distinct from your Production environment—with distinct controls and policies. The Production and Staging environments can be created and governed as separate OUs, and billed as separate accounts. In addition, you may want to set up a Sandbox OU for code testing.

Use a well-planned structure for OUs in your landing zone

AWS Control Tower sets up some OUs for you automatically. As your workloads and requirements expand over time, you can extend the original landing zone configuration to suit your needs.

> **Note**
>
> The names given in the examples follow the suggested AWS naming conventions for setting up a multi-account AWS environment. You can rename your OUs after you've set up your landing zone, by selecting **Edit** on the OU detail page.

Recommendations

After AWS Control Tower sets up the first, required OU for you — the Security OU — we recommend creating some additional OUs in your landing zone.

We recommend that you allow AWS Control Tower to create at least one additional OU, called the Sandbox OU. This OU is for your software development environments. AWS Control Tower can set up the Sandbox OU for you during landing zone creation, if you select it.

Two recommended other OUs you can set up on your own: the Infrastructure OU, to contain your shared services and networking accounts, and an OU to contain your production workloads, called the Workloads OU. You can add additional OUs in your landing zone through the AWS Control Tower console on the **Organizational units** page.

Recommended OUs besides the ones set up automatically

- **Infrastructure OU** – Contains your shared services and networking accounts.

  > **Note**
  >
  > AWS Control Tower does not set up the Infrastructure OU for you.

- **Sandbox OU** – A software development OU. For example, it may have a fixed spending limit, or it may not be connected to the production network.

  > **Note**
  >
  > AWS Control Tower recommends that you set up the Sandbox OU, but it is optional. It can be set up automatically as part of configuring your landing zone.

- **Workloads OU** – Contains accounts that run your workloads.

> **Note**
>
> AWS Control Tower does not set up the Workloads OU for you.

For more information see Production starter organization with AWS Control Tower.

Example of AWS Control Tower with a complete multi-account OU structure

AWS Control Tower supports a nested OU hierarchy, which means that you can create a hierarchical OU structure that meets your organization's requirements. You can build an AWS Control Tower environment to match the AWS multi-account strategy guidance.

You also can build a simpler, flat OU structure that performs well and aligns with the AWS multi-account guidance. Just because you can build a hierarchical OU structure, it does not mean that you must do so.

- To view a diagram that shows an example set of OUs in an expanded, flat AWS Control Tower environment with AWS multi-account guidance, see Example: Workloads in a Flat OU Structure.
- For more information about how AWS Control Tower works with nested OU structures, see Nested OUs in AWS Control Tower.
- For more information about how AWS Control Tower aligns with the AWS guidance, see the AWS white paper, Organizing Your AWS Environment Using Multiple Accounts.

The diagram on the linked page shows that more Foundational OUs and more Additional OUs have been created. These OUs serve the additional needs of a larger deployment.

In the Foundational OUs column, two OUs have been added to the basic structure:

- **Security_Prod OU** – Provides a read-only area for security policies, as well as a break-glass security audit area.
- **Infrastructure OU** – You may wish to separate the Infrastructure OU, recommended previously, into two OUs, Infrastructure_Test (for pre-production infrastructure) and Infrastructure_Prod (for production infrastructure).

In the Additional OUs area, several more OUs have been added to the basic structure. These following are the next recommended OUs to create as your environment grows:

- **Workloads OU** – The Workloads OU, recommended previously but optional, has been separated into two OUs, Workloads_Test (for pre-production workloads) and Workloads_Prod (for production workloads).
- **PolicyStaging OU** – Allows system administrators to test their changes to controls and policies before fully applying them.
- **Suspended OU** – Offers a location for accounts that may have been disabled temporarily.

About the Root

The Root is not an OU. It is a container for the management account, and for all OUs and accounts in your organization. Conceptually, the Root contains all of the OUs. It cannot be deleted. You cannot govern enrolled accounts at the Root level within AWS Control Tower. Instead, govern enrolled accounts within your OUs. For a helpful diagram, see the AWS Organizations documentation.

Administrative tips for landing zone setup

Here are some tips for setting up and configuring your landing zone.

- The AWS Region where you do the most work should be your home Region.
- Set up your landing zone and deploy your Account Factory accounts from within your home Region.
- If you're investing in several AWS Regions, be sure that your cloud resources are in the Region where you'll do most of your cloud administrative work and run your workloads.
- By keeping your workloads and logs in the same AWS Region, you reduce the cost that would be associated with moving and retrieving log information across regions.
- The audit and other Amazon S3 buckets are created in the same AWS Region from which you launch AWS Control Tower. We recommend that you do not move these buckets.
- You can make your own log buckets in the Log Archive account, but it is not recommended. Be sure to leave the buckets created by AWS Control Tower.
- Your Amazon S3 access logs must be in the same AWS Region as the source buckets.

- When launching, AWS Security Token Service (STS) endpoints must be activated in the management account, for all Regions supported by AWS Control Tower. Otherwise, the launch may fail midway through the configuration process.
- *AWS Control Tower supports tagging for enabled controls only.* For more information, see AWS Control Tower control tagging APIs.
- We recommend enabling multi-factor authentication (MFA) for every account that AWS Control Tower manages.
- Alternatively, you can use the AWS Root Access Management feature, which allows root actions to be performed on member accounts, and removes the need to enable MFA for every account. For more information, see Centrally managing root access for customers using AWS Organizations.

Considerations about VPCs

- The VPC created by AWS Control Tower is limited to the AWS Regions in which AWS Control Tower is available. Some customers whose workloads run in non-supported Regions may want to disable the VPC that is created with your Account Factory account. They may prefer to create a new VPC using the Service Catalog portfolio, or to create a custom VPC that runs only in the required Regions.
- The VPC created by AWS Control Tower is not the same as the default VPC that is created for all AWS accounts. In Regions where AWS Control Tower is supported, AWS Control Tower deletes the default VPC when it creates the AWS Control Tower VPC.
- If you delete your default VPC in your home AWS Region, it's best to delete it in all other AWS Regions.

Recommendations for setting up groups, roles, and policies

As you set up your landing zone, it's a good idea to decide ahead of time which users will require access to certain accounts and why. For example, a security account should be accessible only to the security team, the management account should be accessible only to the cloud administrators' team, and so forth.

For more information about this topic, see Identity and access management in AWS Control Tower.

Recommended restrictions

You can restrict the scope of administrative access to your organizations by setting up an IAM role or policy that allows administrators to manage AWS Control Tower actions only. The recommended approach is to use the IAM policy `arn:aws:iam::aws:policy/service-role/AWSControlTowerServiceRolePolicy`. With the `AWSControlTowerServiceRolePolicy` role enabled, an administrator can manage AWS Control Tower only. Be sure to include appropriate access to AWS Organizations for managing your preventive controls, and SCPs, and access to AWS Config, for managing detective controls, in each account.

When you're setting up the shared audit account in your landing zone, we recommend that you assign the `AWSSecurityAuditors` group to any third-party auditors of your accounts. This group gives its members read-only permission. An account must not have write permissions on the environment that it is auditing, because it can violate compliance with Separation of Duty requirements for auditors.

You can impose conditions in your role trust policies, to restrict the accounts and resources that interact with certain roles in AWS Control Tower. We strongly recommend that you restrict access to the `AWSControlTowerAdmin` role, because it allows wide access permissions. for more information, see Optional conditions for your role trust relationships.

Guidance for creating and modifying AWS Control Tower resources

We recommend the following best practices as you create and modify resources in AWS Control Tower. This guidance might change as the service is updated. Remember that the shared responsibility model applies to your AWS Control Tower environment.

General Guidance

- Do not modify or delete any resources created by AWS Control Tower, including resources in the management account, in the shared accounts, and in member accounts. If you modify these resources, you may be required to update your landing zone or re-register an OU, and modification can result in inaccurate compliance reporting.

  In particular:

  - Keep an active AWS Config recorder. If you delete your Config recorder, detective controls cannot detect and report drift. Non-compliant resources may be reported as **Compliant** due to insufficient information.

  - Do not modify or delete the AWS Identity and Access Management (IAM) roles created within the shared accounts in the Security organizational unit (OU). Modification of these roles can require an update to your landing zone.
  - Do not delete the `AWSControlTowerExecution` role from your member accounts, even in unenrolled accounts. If you do, you will not be able to enroll these accounts with AWS Control Tower, or register their immediate parent OUs.
- Do not disallow usage of any AWS Regions through either SCPs or AWS Security Token Service (AWS STS). Doing so will cause AWS Control Tower to enter an undefined state. If you disallow Regions with AWS STS, your functionality will fail in those Regions, because authentication would be unavailable in those Regions. Instead, rely on the AWS Control Tower Region deny capability, as shown in the control, Deny access to AWS based on the requested AWS Region, which works at the landing zone level, or the control Region deny control applied to the OU, which works at the OU level to restrict access to Regions.
- The AWS Organizations `FullAWSAccess` SCP must be applied and should not be merged with other SCPs. Change to this SCP is not reported as drift; however, some changes may affect AWS Control Tower functionality in unpredictable ways, if access to certain resources is denied. For example, if the SCP is detached, or modified, an account may lose access to an AWS Config recorder or create a gap in CloudTrail logging.
- Do not use the AWS Organizations `DisableAWSServiceAccess` API to turn off AWS Control Tower service access to the organization where you've set up your landing zone. If you do so, certain AWS Control Tower drift detection features may not function properly without messaging support from AWS Organizations. These drift detection features help guarantee that AWS Control Tower can report the compliance status of of organizational units, accounts, and controls in your organization accurately. For more information, see API_DisableAWSServiceAccess in the AWS Organizations API Reference.
- In general, AWS Control Tower performs a single action at a time, which must be completed before another action can begin. For example, if you attempt to provision an account while the process of enabling a control is already in operation, account provisioning will fail.

  Exception:

  - AWS Control Tower allows concurrent actions to deploy optional controls. For more information, see Concurrent deployment for optional controls.
  - AWS Control Tower allows up to ten concurrent create, update, or enroll actions on accounts, with Account Factory.

> **Note**
>
> For more information about the resources created by AWS Control Tower, see What are the shared accounts?.

Tips about accounts and OUs

- We recommend that you keep each registered OU to a maximum of 1000 accounts, so that you can update those accounts with the **Re-register OU** capability whenever account updates are required, such as when you configure new Regions for governance.
- To reduce the time required when registering an OU, we recommend that you keep the number of accounts per OU to around 680, even though the limit is 1000 accounts per OU. As a general rule, the time required to register an OU increases according to the number of Regions in which your OU is operating, multiplied by the number of accounts in the OU.
- As an estimate, an OU with 680 accounts may require up to 2 hours to register and enable controls, and up to 1 hour to re-register. Also, an OU that has many controls takes longer to register than an OU with few controls.
- One concern about allowing a longer timeframe for registering an OU is that this process blocks other actions. Some customers are comfortable allowing longer times to register or re-register an OU, because they prefer to allow more accounts in each OU.

When to sign in as a root user

Certain administrative tasks require that you must sign in as a root user. You can sign in as a root user to an AWS account that was created by account factory in AWS Control Tower.

You must sign in as a root user to perform the following actions:

- Change certain account settings, including the account name, root user password, or email address. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.
- To close an AWS account.
- For more information about actions that require root user login credentials, see Tasks that require root user credentials in the *AWS Account Management Reference Guide*.

> **Note**
>
> To change or enable your AWS Support plan, you must be signed in as the root user *or* be a user with the appropriate IAM permissions. .

To sign in as root user

1. Open the AWS sign-in page.

   If you don't have the email address of the AWS account to which you require access, you can get it from AWS Control Tower. Open the console for the management account, choose **Accounts**, and look for the email address.
2. Enter the email address of the AWS account to which you require access, and then choose **Next**.
3. Choose **Forgot password?** to have password reset instructions sent to the root user email address.
4. Open the password reset email message from the root user mailbox, then follow the instructions to reset your password.
5. Open the AWS sign-in page, then sign in with your reset password.

Alternatively, you can use the AWS Root Access Management feature, which allows root actions to be performed on member accounts, without needing to sign in as Root. For more information, see Centrally managing root access for customers using AWS Organizations.

AWS Organizations guidance

AWS Control Tower is closely associated with AWS Organizations. Here is some specific guidance about how they work together best to protect your AWS environment.

- You can find guidance about best practices to protect the security of your AWS Control Tower management account and member accounts in the AWS Organizations documentation.
  - Best practices for the management account
  - Best practices for member accounts
- Do not update existing service control policies (SCPs) attached to an OU that is registered with AWS Control Tower. Doing so could result in the controls entering an unknown state, which will

require you to reset your landing zone or re-register your OU in AWS Control Tower. Instead, you can use AWS Organizations to create new SCPs and attach those to the OUs rather than editing the SCPs that AWS Control Tower has created.

- Moving individual, already enrolled, accounts into AWS Control Tower, from outside of a registered OU, causes drift that must be resolved. See Types of governance drift.
- If you use AWS Organizations to create, invite, or move accounts within an organization registered with AWS Control Tower, those accounts are not enrolled by AWS Control Tower and those changes are not recorded. If you need access to these accounts through SSO, see Member Account Access.
- If you use AWS Organizations to move an OU into an organization created by AWS Control Tower, the external OU is not registered by AWS Control Tower.
- AWS Control Tower handles permission filtering differently than AWS Organizations does. If your accounts are provisioned with AWS Control Tower account factory, end-users can see the names and parents of all OUs in the AWS Control Tower console, even if they don't have permission to retrieve those names and parents from AWS Organizations directly.
- AWS Control Tower does not support mixed permissions on organizations, such as permission to view an OU's parent but not to view OU names. For this reason, AWS Control Tower administrators are expected to have full permissions.
- The AWS Organizations `FullAWSAccess` SCP must be applied and should not be merged with other SCPs. Change to this SCP is not reported as drift; however, some changes may affect AWS Control Tower functionality in unpredictable ways, if access to certain resources is denied. For example, if the SCP is detached, or modified, an account may lose access to an AWS Config recorder or create a gap in CloudTrail logging.
- Don't use the AWS Organizations `DisableAWSServiceAccess` API to turn off AWS Control Tower service access to the organization where you've set up your landing zone. If you do so, certain AWS Control Tower drift detection features may not function properly without messaging support from AWS Organizations. These drift detection features help guarantee that AWS Control Tower can report the compliance status of of organizational units, accounts, and controls in your organization accurately. For more information, see API_DisableAWSServiceAccess in the AWS Organizations API Reference.

IAM Identity Center guidance

AWS Control Tower recommends that you use AWS Identity and Access Management (IAM) to regulate access to your AWS accounts. However, you have the option to choose whether AWS

Control Tower sets up IAM Identity Center for you, whether you set up IAM Identity Center for yourself, in a way that meets your business requirements most effectively, or whether to select another method for account access.

> **Note**
>
> **SSO** is an abbreviation used in the technology industry to denote *single sign-on*. In general terms, SSO is a session and user authentication service. It permits someone to use one set of login credentials for access to many applications. When referring to the single-sign on capability in AWS, we are referring to the AWS service called **AWS Identity and Access Management**, and abbreviated as **IAM** or **IAM Identity Center**.

By default, AWS Control Tower sets up AWS IAM Identity Center for your landing zone, in alignment with best-practices guidance defined in [Organizing your AWS environment using multiple accounts](). Most customers choose the default. Alternative access methods are required sometimes, for regulatory compliance in specific industries or countries, or in AWS Regions where AWS IAM Identity Center is not available.

Choosing an option

From the console, you can choose to self-manage IAM Identity Center during the landing zone set up process, rather than allowing AWS Control Tower to set it up for you. At any time later, you can choose to change this selection, by modifying the landing zone settings and updating your landing zone on the landing zone **Settings** page.

To discontinue AWS IAM Identity Center in AWS Control Tower, or to begin using AWS IAM Identity Center

1. Navigate to the landing zone **Settings** page
2. Select the **Configurations** tab
3. Then choose the appropriate radio button, to change your selection for AWS IAM Identity Center.

After you choose to self-manage AWS IAM Identity Center as your IdP, AWS Control Tower creates only those roles and policies needed to manage AWS Control Tower, such as `AWSControlTowerAdmin` and `AWSControlTowerAdminPolicy`. For landing zones that self-manage, AWS Control Tower no longer creates IAM roles and groupings for customer-specific use — not during the landing zone set-up process, nor during account provisioning with Account Factory.

> **Note**
>
> If you remove AWS IAM Identity Center from your AWS Control Tower landing zone, the users, groups, and permission sets that AWS Control Tower created are not removed. We recommend that you remove these resources.

Account Factory customers with alternative identity providers (IdPs) such as Azure AD, Ping, or Okta, can follow the AWS IAM Identity Center process to connect to an external identity provider and onboard their IdP. You can return to having AWS Control Tower generate your groupings and roles at any time, by modifying the landing zone settings.

- For specific information about how AWS Control Tower works with IAM Identity Center based on your identity source, see **Considerations for AWS IAM Identity Center customers** in the Pre-launch checks section of the *Getting Started* page of this User Guide.
- For additional information about how the behavior of AWS Control Tower interacts with IAM Identity Center and different identity sources, refer to Considerations for Changing Your Identity Source in the *IAM Identity Center User Guide*.
- See Working with AWS IAM Identity Center and AWS Control Tower for more information about working with AWS Control Tower and IAM Identity Center.

Account Factory guidance

You can encounter issues when using Account Factory to provision a new account in AWS Control Tower. For information about how to troubleshoot these issues, see the section New Account Provisioning Failed in Troubleshooting of the *AWS Control Tower User Guide*.

We recommend that you create federated users or IAM roles instead of IAM users. Federated users and IAM roles provide you with temporary credentials. IAM users have long-term credentials that can be difficult to manage. For more information, see IAM identities (users, user groups, and roles) in the *IAM User Guide*.

If you're authenticated as an IAM user or IAM Identity Center user when provisioning a new account in Account Factory or when using the *Enroll account* feature AWS Control Tower, verify that your user has access to your AWS Service Catalog portfolio. Otherwise, you might receive an error message from Service Catalog. For more information, see No Launch Paths Found Error in the Troubleshooting section of the *AWS Control Tower User Guide*.

> **Note**
>
> Up to five accounts can be provisioned at a time.

Guidance on subscribing to SNS Topics

Subscribe to SNS topics to get information about your AWS Control Tower environment.

- The `aws-controltower-AllConfigNotifications` SNS topic receives all events published by AWS Config, including compliance notifications and Amazon CloudWatch event notifications. For example, this topic informs you if a control violation has occurred. It also gives information about other types of events. (Learn more from AWS Config about what they publish when this topic is configured.)
- Data Events from the `aws-controltower-BaselineCloudTrail` trail are set to publish to the `aws-controltower-AllConfigNotifications` SNS topic as well.
- To receive detailed compliance notifications, we recommend that you subscribe to the `aws-controltower-AllConfigNotifications` SNS topic. This topic aggregates compliance notifications from all child accounts.
- To receive drift notifications and other notifications as well as compliance notifications, but fewer notifications overall, we recommend that you subscribe to the `aws-controltower-AggregateSecurityNotifications` SNS topic.
- To receive notifications about AWS Control Tower Account Factory for Terraform (AFT) errors, you can subscribe to an SNS topic called aft_failure_notifications, shown in the AFT repository. For example:

```
resource "aws_sns_topic" "aft_failure_notifications" {
    name = "aft-failure-notifications"
    kms_master_key_id = "alias/aws/sns"
}
```

- All SNS topics are encrypted at rest with disk encryption. for more information, see Data encryption.

For more information about SNS topics and compliance, see Prevention and notification.

Guidance for KMS keys

AWS Control Tower works with AWS Key Management Service (AWS KMS). Optionally, if you wish to encrypt and decrypt your AWS Control Tower resources with an encryption key that you manage, you can generate and configure AWS KMS keys. You can add or change a KMS key any time you update your landing zone. As a best practice, we recommend using your own KMS keys and changing them from time to time.

AWS KMS allows you to create multi-Region KMS keys and asymmetric keys. However, AWS Control Tower does not support multi-Region keys or asymmetric keys. AWS Control Tower performs a pre-check of your existing keys. You may see an error message if you select a multi-Region key or an asymmetric key. In that case, generate another key for use with AWS Control Tower resources.

For customers who operate an AWS CloudHSM cluster: Create a custom key store associated with your CloudHSM cluster. Then you can create a KMS key, which resides in the CloudHSM custom key store you created. You can add this KMS key to AWS Control Tower.

You must make a specific update to the permissions policy of a KMS key to make it work with AWS Control Tower. For details, refer to the section called Update the KMS key policy.

Best practices for landing zone updates

This section gives some considerations and best practices to keep in mind when you are considering an upgrade of your landing zone version in AWS Control Tower. The change from the 2.0 landing zone version series into the 3.0 landing zone version series is especially important. When you upgrade your landing zone, AWS Control Tower automatically moves you to the latest available version.

> **Note**
>
> It is a best practice to update to the latest version of the landing zone.

Summary of best practices explained in this section

- **Best practice:** For security and audit reasons, we strongly recommend that you enable logging across the board, for all accounts, and send logging information to a centralized location. In AWS

Control Tower, this centralized location is the **Log archive** account, which provides an Amazon S3 logging bucket.

- **Best practice:** If you opt out of the organization-level CloudTrail trail in AWS Control Tower, set up and manage your own trails.
- **Best practice:** When operating your AWS Control Tower environment, set up a testing environment.

Benefits for moving from 2.x landing zone versions to 3.x landing zone versions

- Record AWS Config resources only in the home Region, which creates cost savings when you manage global resources
- Encrypt your AWS CloudTrail trail with your own KMS key
- Customize your log retention timeframe
- Enhanced mandatory controls
- Increased number of controls available
- Integrated with AWS Security Hub
- Python runtime updates

Caveats for moving from 2.x landing zone versions to 3.x landing zone versions

- With landing zone 3.0 and later, AWS Control Tower no longer supports account-level AWS CloudTrail trails that AWS manages.
- You have an option to choose an organization-level trail managed by AWS Control Tower, or to opt out of it and manage your own CloudTrail trails.
- Some potential exists for double costs, especially if some accounts within an OU are not enrolled in AWS Control Tower and have account-level trails of their own that you want to keep.

Considerations about choosing organization-level CloudTrail trails

- When you upgrade to 3.0 or later, AWS Control Tower deletes the account-level trails that it originally created, after 24 hours. [Exception]
- No data from these trails is lost. Your existing logs are preserved even when the trails are removed.

- AWS Control Tower creates a new path in the same Amazon S3 bucket for the trails, to differentiate account-level trails from organization-level trails.
  - An account trail log path is of this form: `/orgId/AWSLogs/...`
  - An organization trail log path is of this form: `/orgId/AWSLogs/orgId/...`
- Additional CloudTrail trails that you have deployed, trails not deployed by AWS Control Tower, are not touched.
- All accounts are included in the organization-level trail—including accounts not enrolled in AWS Control Tower—if the unenrolled accounts are part of a registered OU.
- Amazon CloudWatch alarms in linked accounts are not triggered.
- If you opt out of an organization-level trail, AWS Control Tower still creates the trail, but sets its status to **Off**.
- As a best practice, if you opt out of the organization-level trail in AWS Control Tower, you should set up and manage your own CloudTrail trails,

Benefits of organization-level trails

- The organization trail works across all accounts in the OU.
- The logged items are standardized and cannot be modified by account users.

Consider a testing environment

When you upgrade your landing zone, AWS Control Tower makes changes only to the shared accounts and the Foundational OU. It does not make changes to your workload accounts or OUs. *However, as a best practice, when operating your AWS Control Tower environment, we recommend that you set up a testing environment.* Within the isolated testing environment, you can test the AWS Control Tower landing zone upgrades, as well as any changes you may make to service control policies (SCPs), and you can test the controls that you wish to apply to the environment. This recommendation is especially helpful if you are operating in a regulated industry.

Checklist for common errors when updating

Here is a short list of tasks you can perform to avoid common errors when updating your AWS Control Tower landing zone from 2.x versions to 3.x versions.

Basic update checklist

- Check your landing zone:

– Go to the AWS Control Tower service, review the **Organizational units** and **Accounts** pages, and then confirm that your account state is set to **Registered** and **Enrolled**.

– If applicable, verify and confirm that the last run of your customizations pipeline was successful.

– Check the Amazon S3 centralized logging bucket in the **Audit** account, because any changes previously made to the bucket policy will be overwritten.

- Validate that any SCPs not owned by AWS Control Tower will not restrict the `AWSControlTowerExecution` role from performing actions in member accounts, or actions in the management account, for the administrative role that's performing the update.

AI-based services and AWS Control Tower

You can create service control policies (SCPs) that allow you to opt out of having your data stored by AI-based services on AWS. These SCP policies specify that AI-based services, such as Amazon Rekognition or Amazon CodeWhisperer, cannot store and use your data to improve other AI-based AWS services.

These AI opt-out SCP policies can apply to your entire organization, to an OU, or to a specific account. The policies are global in effect. You can find more information about these policies at AI services opt-out policies, in the AWS Organizations documentation.

For a list of AWS services that use AI, along with examples of policies, see AI services opt-out policy syntax and examples, in the *AWS Organizations User Guide*.

Configuration update management in AWS Control Tower

It is the responsibility of the members of your central cloud administrators' team to keep your landing zone updated. Updating your landing zone ensures that AWS Control Tower is patched and updated. In addition, to protect your landing zone from potential compliance issues, the members of the central cloud administrator team should resolve drift issues as soon as they're detected and reported.

Note

The AWS Control Tower console indicates when your landing zone needs to be updated. If you don't see an option to update, your landing zone is already up to date.

The following table contains a list of AWS Control Tower landing zone update releases, with links to descriptions of each release.

Version	Release date	Description
3.3	12-12-2023	Landing zone version 3.3
3.2	6-09-2023	Landing zone version 3.2
3.1	2-09-2023	Landing zone version 3.1
3.0	7-26-2022	Landing zone version 3.0
2.9	4-22-2022	Landing zone version 2.9
2.8	2-10-2022	Landing zone version 2.8
2.7	4-8-2021	Landing zone version 2.7
2.6	12-29-2020	Landing zone version 2.6
2.5	11-18-2020	Landing zone version 2.5

Version	Release date	Description
2.4	None	None
2.3	3-5-2020	Landing zone version 2.3
2.2	11-13-19	Landing zone version 2.2
2.1	6-24-19	Landing zone version 2.1

Each time you update your landing zone, you have the opportunity to modify your landing zone settings.

Benefits of updating

- You can change your governed Regions
- You can change your log retention policy
- You can add or remove the Region deny control
- You can apply AWS KMS encryption keys
- You can activate or deactivate your organization-level CloudTrail trail.
- You can resolve landing zone drift

When you update your landing zone, you receive the latest features for AWS Control Tower, automatically. View your current landing zone version on the **Landing zone settings** page.

If an update fails, AWS Control Tower does not roll back to a previous landing zone version. You may find your landing zone in an indeterminate state. If so, contact AWS support. For more information about troubleshooting a failure to update, see Unable to Update Landing Zone.

You have the opportunity to clear unused AWS Identity center (formerly called AWS SSO) mappings when you update your landing zone. For more information, see Field Notes: Clear Unused IAM Identity Center Mappings Automatically During AWS Control Tower Upgrades.

Prerequisite for Update and Reset – turn off Requester Pays

Before you update or reset your landing zone, be sure that the Amazon S3 logging bucket for the Log Archive account does not have the **Requester Pays** feature enabled. You must turn off that feature before you begin the **Update** or **Reset** process. When AWS Control Tower sets up your logging bucket, this feature is not enabled. Therefore, only the customers who have subesquently activated the Requester Pays feature must turn it off. For more information, see Amazon S3 bucket policy for CloudTrail and Using Requester Pays buckets.

About landing zone updates

Updates are required to correct governance drift, or to move to a new version of AWS Control Tower. To perform a complete update of AWS Control Tower, you must update your landing zone first and then update the enrolled accounts individually. You may need to perform three types of updates at different times.

- **A landing zone update:** Most often this type of update is performed by choosing **Update** on the **Landing zone settings** page. You may need to perform a landing zone update to resolve certain types of drift, and you can choose **Reset** when necessary.
- **An update of one or more individual accounts:** You must update accounts if the associated information changes, or if certain types of drift have occurred. If an account requires an update, the account's status will show **Update available** on the **Accounts** page.

  To update a single account, navigate to the account detail page and select **Update account**. Accounts also may be updated by a manual process, by choosing **Re-register OU**, or with an automated scripting approach, described in a later section of this page.
- **A full update:** A full update includes an update of your landing zone, followed by an update of all the enrolled accounts in your registered OUs. Full updates are required with a new release of AWS Control Tower such as 3.0, 3.2, and so forth. To make the full update process easier, for OUs with 1000 or fewer accounts, you can choose **Re-register OU** to update all of the accounts within that OU, and repeat the **Re-register OU** command for each OU.

For more information about landing zone updates, see Best practices for landing zone updates.

Note

After completing a landing zone update, you cannot undo the update or downgrade to a previous version.

Update your landing zone

The easiest way to update your AWS Control Tower landing zone is through the **Landing zone settings** page, which you can reach by choosing **Landing zone settings** in the left navigation of the AWS Control Tower dashboard.

The **Landing zone settings** page shows you the current version of your landing zone, and it lists any updated versions that may be available. You can choose the **Update** button if you need to update your version.

> **Note**
>
> Alternatively, you can update your landing zone manually. The update takes approximately the same amount of time, whether you use the **Update** button or the manual process. To perform a manual update of your landing zone only, see steps 1 and 2 that follow.

Standard update procedure

The following procedure walks you through the steps of a full update for AWS Control Tower from the console. To update an individual account, see Update the account in the console.

To update your landing zone, with any number of accounts per OU

1. Open a web browser, and navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower/home/update.
2. Review the information in the wizard and choose **Update**. This updates the backend of the landing zone as well as your shared accounts. This process can take a little more than half an hour.
3. Update your member accounts (this procedure must be followed for an OU that contains over 1000 accounts).
4. From the left navigation pane, choose **Organization**.
5. To update each account, follow the steps given in Update the account in the console.

Optionally Re-register OU to update accounts

For registered AWS Control Tower OUs with fewer than 1000 accounts, you can go to the **OU page** in the dashboard and select **Re-register OU** to update the accounts in that OU.

Select a landing zone version

If you are running AWS Control Tower landing zone version 3.1 and above, you can choose to stay on the current version, or you can upgrade to a newer version, when you perform an **Update** or **Reset** operation on your landing zone configurations. The **Reset** operation is the best way to repair drift, in most situations.

You can choose a landing zone version in the AWS Control Tower console, or by means of the AWS Control Tower APIs.

Note

If you choose to deploy a landing zone version that skips over an intermediate version, for example if you move from 3.1 to 3.3, AWS Control Tower automatically deploys the intermediate version as part of the update operation.
In conversation, moving to a newer version is often referred to as an *upgrade*, not just an update. These two concepts are distinct, because you can *update* your landing zone settings without upgrading to a new version, for example, by changing the Regions that you govern.
In the console, the **Update** button performs an in-place update or an upgrade operation, based on your current landing zone version and the one you select to deploy.

Choose your landing zone version – console procedure

1. From the AWS Control Tower console, navigate to the **Landing zone settings** page. In the table of available landing zones, select the new version. Remember that you can select versions 3.1 or later. Versions previous to 3.1 are not compatible with this feature.
2. When you select a version from the table, you can see the available actions. **Update** is available if your current version is earlier than the selected version. **Reset** is available if your current version is 3.1 or newer.
3. After you choose the version, select the **Update** button or the **Reset** button, in the upper right area of the screen.

4. You will see a confirmation display showing the landing zone version that you've selected for deployment. To continue, choose **Next** at the lower right. Your update operation may take a few minutes or more.
5. After the landing zone is updated, you may need to update your accounts. The easiest way to do the account updates is by a **Re-register OU** process for each of your registered OUs.

Account updates, landing zone versions, and baselines

AWS Control Tower landing zones are AWS resources that correspond to a set of baseline configurations. There is not a one-to-one mapping of baselines and landing zone versions. You can view a table that shows Compatibility of OU baselines and landing zone versions.

When you jump a baseline version, you must update accounts after your landing zone update. For example, when upgrading from 3.1 to 3.2, you would not need to update your accounts, because these landing zone versions share the same baseline.

In contrast, if you upgrade from 3.1 to 3.3, you would have to update accounts, because the baseline version is 4.0, which encompasses 3.2 to 3.3.

For more information about the relationship between landing zone versions and baselines, see Compatibility of OU baselines and landing zone versions.

Retain AWS CloudTrail trails during landing zone update

You can choose to retain your account-level AWS CloudTrail trails when you upgrade your AWS Control Tower landing zone version.

Prerequisites

- Your landing zone version is less than 3.0.
- Your most recent **Create** or **Update** operation succeeded.

To retain the account-level trail and opt in to organization-level CloudTrail trails

1. Contact AWS Support with a request to allowlist your account.
2. The support team confirms when the target account is allowlisted.
3. After confirmation, update your landing zone to version 3.1 or greater, and choose **AWS CloudTrail configuration - Enabled**.

To retain the account-level trail and opt out of CloudTrail trails managed by AWS Control Tower

1. Contact AWS Support with a request to allowlist your account.
2. The support team confirms when the target account is allowlisted.
3. After confirmation, update your landing zone to version 3.1 or greater and choose **AWS CloudTrail configuration - Not Enabled**.

> **Important**
>
> After the account-level CloudTrail trails are retained, we cannot remove trails or remove your accounts from the allow list.

How to make a support request to retain your account-level trails

If you need to retain account-level trails during a Landing Zone update, you must contact AWS Support to add your account to the AWS Control Tower allow list. Follow these steps to submit a support ticket:

1. Sign in to the AWS Management Console.
2. Navigate to the AWS Support Center.
3. Choose **Create case**.
4. For **Case type**, select **Technical support**.
5. For **Service**, choose **AWS Control Tower**.
6. For **Category**, select **General Guidance**.
7. In the **Subject** line, include the following phrase:

   ```
   Allow retention of account-level trails during Landing Zone update
   ```
8. In the **Description** field, provide the following details:
   - Your AWS Management account number
   - The selected home Region for your AWS Control Tower environment
9. Complete any other required fields in the support case form.
10. Choose **Submit** to create the support case.

After you submit the ticket, AWS Support reviews your request and adds your account to the allow list, if appropriate. You will receive further instructions and confirmation through the support case communication channel.

> **Note**
>
> To delete the account-level trail after it is allowlisted, use the management account to delete the AWS CloudFormation stack set or specific stack instance. All resources in the stack are deleted.

Resolve drift with Reset and Re-register

Drift often occurs as you and your organization members use the landing zone.

Drift detection is automatic in AWS Control Tower. Automated scans of your SCPs help you identify resources that need changes or configuration updates that must be made to resolve the drift.

To repair many types of drift, choose **Reset** on the **Landing zone settings** page in the console. Also, you can resolve some types of drift by choosing to **Re-register** an OU in the console. For controls, you can resolve drift programmatically by calling the **ResetEnabledControl** API. For more information about types of drift and how to resolve them, see Types of governance drift and Detect and resolve drift in AWS Control Tower.

One special case of drift resolution occurs for *role drift*. If a required role is not available, the console shows a warning page and some instructions on how to restore the role. Your landing zone is unavailable until the role drift is resolved. This drift reset is not the same as a full landing zone reset. For more information, see *Don't delete required roles* in the section called Types of drift to resolve right away.

> **When you take action to resolve drift on a landing zone version, two behaviors are possible.**
>
> - If you are on the latest landing zone version, when you choose **Reset** and then choose **Confirm**, your drifted landing zone resources are reset to the saved AWS Control Tower configuration. The landing zone version stays the same.
> - If you are not on the latest version, you must choose **Update**. The landing zone is upgraded to the latest landing zone version. Drift is resolved as part of this process.

Provision and update accounts using automation

You can provision or update individual accounts in AWS Control Tower by several methods:

- You can provision and customize accounts with *AWS Control Tower Account Factory for Terraform* (AFT). For more information, see Overview of AWS Control Tower Account Factory for Terraform (AFT).
- You can update accounts with *Customizations for AWS Control Tower* (CfCT). For more information, see Customizations for AWS Control Tower (CfCT) overview .
- **Script automation**: If you prefer to use an API approach, you can update accounts using the API framework of Service Catalog and the AWS CLI to update the accounts in a batch process. You'd call the `UpdateProvisionedProduct` API of Service Catalog for each account. You can write a script to update the accounts, one by one, with this API. More information about this approach, when adding Regions for governance, is available in a blog post, Enabling guardrails in new AWS Regions.

  You can update as many as five (5) accounts at a time. You must wait for at least one account update to succeed before beginning the next account update. Therefore, the process may take a long time if you have a lot of accounts, but it is not complicated. For more information about this approach, see the Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs.

> **Video walkthrough**
>
> The Video Walkthrough is designed for automated account provisioning with a script, but the steps also apply to account updating. Use the `UpdateProvisionedProduct` API instead of the `ProvisionProduct` API.

A further step of automation by script is to check for **Succeed** status of the AWS Control Tower `UpdateLandingZone` lifecycle event. Use it as a trigger to begin updating individual accounts as described in the video. A lifecycle event marks the completion of a sequence of activities, so the occurrence of this event means that a landing zone update is complete. The landing zone update must be complete before account updates begin. For more information about working with lifecycle events, see Lifecycle Events.

Also see:

- Use AWS CloudShell to work with AWS Control Tower.
- Automate tasks in AWS Control Tower .

Automate tasks in AWS Control Tower

Many customers prefer to automate tasks in AWS Control Tower, such as account provisioning, control assignment, and auditing. You can set up these automated actions with calls to:

- AWS Service Catalog APIs
- AWS Organizations APIs
- AWS Control Tower APIs
- the AWS CLI

The Additional information and links page contains links to many excellent technical blog posts that can help you automate tasks in AWS Control Tower. The sections that follow provide links to areas in this *AWS Control Tower User Guide* that can assist you with automating tasks.

Automating control tasks

You can automate tasks related to applying and removing controls (also known as *guardrails*) through the AWS Control Tower API. For details, see the AWS Control Tower API Reference.

For more information about how to perform control operations with AWS Control Tower APIs, see the blog post AWS Control Tower releases API, pre-defined controls to your organizational units.

Automating landing zone tasks

The AWS Control Tower landing zone APIs help you automate certain tasks related to your landing zone. For details, see the AWS Control Tower API Reference.

Automating OU registration

The AWS Control Tower baseline APIs help you automate certain tasks, such as registering an OU. For details, see the AWS Control Tower API Reference.

Automated account closure

You can automate the closure of AWS Control Tower member accounts with an AWS Organizations API. For more information, see Close an AWS Control Tower member account through AWS Organizations.

Automated account provisioning and updating

AWS Control Tower Account Factory Customization (AFC) helps you create accounts from the AWS Control Tower console, with customized AWS CloudFormation templates that we refer to as blueprints. This process is automated in the sense that you can create new accounts and update accounts repeatedly, after setting up a single blueprint, without maintaining pipelines.

AWS Control Tower Account Factory for Terraform (AFT) follows a GitOps model to automate the processes of account provisioning and account updating in AWS Control Tower. For more information, see Provision accounts with AWS Control Tower Account Factory for Terraform (AFT) .

Customizations for AWS Control Tower (CfCT) helps you customize your AWS Control Tower landing zone and stay aligned with AWS best practices. Customizations are implemented with AWS CloudFormation templates, service control policies (SCPs), and resource control policies (RCPs). For more information, see Customizations for AWS Control Tower (CfCT) overview .

For more information and a video about automated account provisioning, see Walkthrough: Automated account provisioning in AWS Control Tower and Automated provisioning with IAM roles.

Also see Update accounts by script.

Programmatic auditing of accounts

For more information about auditing accounts programmatically, see Programmatic roles and trust relationships for the AWS Control Tower audit account.

Automating other tasks

For information about how to increase certain AWS Control Tower service quotas with an automated request method, view this video: Automate Service Limit Increases.

For technical blogs that cover automation and integration use cases, see Automation and integration.

Two open source samples are available on GitHub to help you with certain automation tasks related to security.

- The sample called aws-control-tower-org-setup-sample shows how to automate setting up the Audit account as the delegated administrator for security-related services.
- The sample called aws-control-tower-account-setup-using-step-functions shows how to automate security best practices using Step Functions, when provisioning and configuring new accounts. This sample includes adding principals to organizationally-shared AWS Service Catalog

portfolios and associating organization-wide AWS IAM Identity Center groups to new accounts automatically. It also illustrates how to delete the default VPC in every Region.

The *AWS Security Reference Architecture* includes code examples for automating tasks related to AWS Control Tower. For more information, see the AWS Prescriptive Guidance pages and the associated GitHub repository.

For information about using AWS Control Tower with AWS CloudShell, an AWS service that facilitates working in the AWS CLI, see AWS CloudShell and the AWS CLI.

Because AWS Control Tower is an orchestration layer for AWS Organizations, many other AWS services are available by means of APIs and the AWS CLI. For more information, see Related AWS services.

Use AWS CloudShell to work with AWS Control Tower

AWS CloudShell is an AWS service that facilitates working in the AWS CLI — it's a browser-based, pre-authenticated shell that you can launch directly from the AWS Management Console. There's no need to download or install command line tools. You can run AWS CLI commands for AWS Control Tower and other AWS services from your preferred shell (Bash, PowerShell or Z shell).

When you launch AWS CloudShell from the AWS Management Console, the AWS credentials you used to sign in to the console are available in a new shell session. You can skip entering your configuring credentials when you interact with AWS Control Tower and other AWS services, and you'll be using AWS CLI version 2, which is pre-installed on the shell's compute environment.You're pre-authenticated with AWS CloudShell.

Obtain IAM permissions for AWS CloudShell

AWS Identity and Access Management provides access management resources that allow administrators to grant permissions to IAM users and IAM Identity Center users for access to AWS CloudShell.

The quickest way for an administrator to grant access to users is through an AWS managed policy. An AWS managed policy is a standalone policy that's created and administered by AWS. The following AWS managed policy for CloudShell can be attached to IAM identities:

- `AWSCloudShellFullAccess`: Grants permission to use AWS CloudShell with full access to all features.

If you want to limit the scope of actions that an IAM user or IAM Identity Center user can perform with AWS CloudShell, you can create a custom policy that uses the `AWSCloudShellFullAccess` managed policy as a template. For more information about limiting the actions that are available to users in CloudShell, see [Managing AWS CloudShell access and usage with IAM policies] in the *AWS CloudShell User Guide*.

Note

Your IAM identity also requires a policy that grants permission to make calls to AWS Control Tower. For more information, see [Permissions required to use the AWS Control Tower console].

Launch AWS CloudShell

From the AWS Management Console, you can launch CloudShell by choosing the following options available on the navigation bar:

- Choose the CloudShell icon.
- Start typing "cloudshell" in Search box and then choose the CloudShell option.

Now that you've started CloudShell, you can enter any AWS CLI commands you require to work with AWS Control Tower. For example, you can check your AWS Config status.

Interact with AWS Control Tower through AWS CloudShell

After you launch AWS CloudShell from the AWS Management Console, you can immediately start to interact with AWS Control Tower from the command line interface. AWS CLI commands work in the standard way in CloudShell.

Note

When using AWS CLI in AWS CloudShell, you don't need to download or install any additional resources. You're already authenticated within the shell, so you don't need to configure credentials before making calls.

Use AWS CloudShell to help set up AWS Control Tower

Before performing these procedures, unless it's otherwise indicated, you must be signed in to the AWS Management Console in the home Region for your landing zone, and you must be signed in as an IAM Identity Center user or IAM user with administrative permissions for the management account that contains your landing zone.

1. Here's how you can use AWS Config CLI commands in AWS CloudShell to determine the status of your configuration recorder and delivery channel before you start to configure your AWS Control Tower landing zone.

   Example: Check your AWS Config status

   View commands:

   - `aws configservice describe-delivery-channels`
   - `aws configservice describe-delivery-channel-status`
   - `aws configservice describe-configuration-recorders`
   - The normal response is something like `"name": "default"`

2. If you have an existing AWS Config recorder or delivery channel that you need to delete before you set up your AWS Control Tower landing zone, here are some commands you can enter:

   Example: Manage your pre-existing AWS Config resources

   Delete commands:

   - `aws configservice stop-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
   - `aws configservice delete-delivery-channel --delivery-channel-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
   - `aws configservice delete-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*

   > **Important**
   >
   > Do not delete the AWS Control Tower resources for AWS Config. Loss of these resources can cause AWS Control Tower to enter an inconsistent state.

For more information, see the AWS Config documentation

- Managing the Configuration Recorder (AWS CLI)
- Managing the Delivery Channel

3. This example shows AWS CLI commands you'd enter from AWS CloudShell to enable or disable trusted access for AWS Organizations. For AWS Control Tower you do not need to enable or disable trusted access for AWS Organizations, it is just an example. However, you may need to enable or disable trusted access for other AWS services if you're automating or customizing actions in AWS Control Tower.

   Example: Enable or disable trusted service access

   - `aws organizations enable-aws-service-access`
   - `aws organizations disable-aws-service-access`

Example: Create an Amazon S3 bucket with AWS CloudShell

In the following example, you can use AWS CloudShell to create an Amazon S3 bucket and then use the **PutObject** method to add a code file as an object in that bucket.

1. To create a bucket in a specified AWS Region, enter the following command in the CloudShell command line:

```
aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
```

   If the call is successful, the command line displays a response from the service similar to the following output:

```
{
    "Location": "/insert-unique-bucket-name-here"
}
```

> **Note**
>
> If you don't adhere to the [rules for naming buckets](#) (using only lowercase letters, for example), the following error is displayed: An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.

2. To upload a file and add it as an object to the bucket that was just created, call the **PutObject** method:

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body
 add_prog.py
```

If the object is uploaded successfully to the Amazon S3 bucket, the command line displays a response from the service similar to the following output:

```
{
           "ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebeea56\""}
```

The `ETag` is the hash of the object that's been stored. It can be used to check the integrity of the object uploaded to Amazon S3.

Create AWS Control Tower resources with AWS CloudFormation

AWS Control Tower is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want, such as `AWS::ControlTower::EnabledControl` for controls. AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your AWS Control Tower resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

AWS Control Tower and AWS CloudFormation templates

To provision and configure resources for AWS Control Tower and related services, you must understand AWS CloudFormation templates. Templates are formatted text files in JSON or YAML.

These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

AWS Control Tower supports creating `AWS::ControlTower::EnabledControl` (control resources), `AWS::ControlTower::LandingZone` (landing zones), and `AWS::ControlTower::EnabledBaseline` (baselines) in AWS CloudFormation. For more information, including examples of JSON and YAML templates for these resource types, see AWS Control Tower in the *AWS CloudFormation User Guide*.

> **Note**
>
> The limit for `EnableControl`and `DisableControl` updates in AWS Control Tower is 100 concurrent operations.

To view some AWS Control Tower examples for the CLI and the console, see Enable controls with AWS CloudFormation.

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

Customize your AWS Control Tower landing zone

Certain aspects of your AWS Control Tower landing zone are configurable in the console, such as selection of Regions and optional controls. Other changes may be made outside the console, with automation.

For example, you can create more extensive customizations of your landing zone with the *Customizations for AWS Control Tower* capability, a GitOps-style customization framework that works with AWS CloudFormation templates and AWS Control Tower lifecycle events.

Customize from the AWS Control Tower console

To make these customizations to your landing zone, follow the steps given by the AWS Control Tower console.

Select customized names during setup

- You can select your top-level OU names during setup. You can rename your OUs at any time using the AWS Organizations console, but making changes to your OUs in AWS Organizations may cause repairable drift.
- You can select the names of your shared **Audit** and **Log Archive** accounts, but you cannot change the names after setup. (This is a one-time selection.)

> **Tip**
>
> Remember that renaming an OU in AWS Organizations does not update the corresponding provisioned product in Account Factory. To update the provisioned product automatically (and avoid drift), you must perform the OU operation through AWS Control Tower, including creating, deleting, or re-registering an OU.

Select AWS Regions

- You can customize your landing zone by selecting specific AWS Regions for governance. Follow the steps in the AWS Control Tower console.

- You can select and de-select AWS Regions for governance when you update your landing zone.
- You can set the Region Deny control to **Enabled** or **Not enabled**, and control user access to most AWS services in ungoverned AWS Regions.

For information about AWS Regions where CfCT has deployment limitations, see Control limitations.

Customize by adding optional controls

- Strongly recommended and elective controls are optional, which means that you can customize the level of enforcement for your landing zone by choosing which ones to enable. Optional controls are not enabled by default.
- The optional Data residency controls allow you to customize the Regions in which you store and allow access to your data.
- The optional controls that are part of the integrated Security Hub standard allow you to scan your AWS Control Tower environment to check for security risks.
- The optional proactive controls allow you to check your AWS CloudFormation resources before they are provisioned, to make sure the new resources will comply with your environment's control objectives.

Customize your AWS CloudTrail trails

- When you update your landing zone to version 3.0 or later, you can choose to opt into or opt out of organization-level CloudTrail trails managed by AWS Control Tower. You can change this selection any time you update your landing zone. AWS Control Tower creates an organization-level trail in your management account, and that trail enters active or inactive status, based on your choice. Landing zone 3.0 does not support account-level CloudTrail trails; however, if you require these, you can configure and manage your own trails. You may incur additional cost for duplicate trails.

Create customized member accounts in the console

- You can create AWS Control Tower member accounts that are customized, and you can update existing member accounts to add customizations, from the AWS Control Tower console. For more information, see Customize accounts with Account Factory Customization (AFC).

Automate customizations outside the AWS Control Tower console

Some customizations are not available through the AWS Control Tower console, but they can be implemented in other ways. For example:

- You can customize accounts during provisioning, in a GitOps-style workflow, with *Account Factory for Terraform (AFT)*.

  AFT is deployed with a Terraform module, available in the AFT repository.
- You can customize your AWS Control Tower landing zone with *Customizations for AWS Control Tower* (CfCT), a package of functionality that is built upon AWS CloudFormation templates and service control policies (SCPs). You can deploy the custom templates and policies to individual accounts and organizational units (OUs) within your organization.

  Source code for CfCT is available in a GitHub repository.
- You can customize your AWS Control Tower landing zone with Landing Zone Accelerator (LZA) on AWS. The LZA solution is architected to align with AWS best practices and conform to multiple global compliance frameworks. We recommend that you deploy AWS Control Tower as the foundational landing zone, and then enhance the landing zone capabilities with LZA, as needed. For more information, see AWS Control Tower and Landing zone accelerator.

AWS Control Tower and Landing zone accelerator

This section describes the benefits of working with AWS Control Tower and the **Landing Zone Accelerator** (LZA) solution, together.

You can customize your AWS Control Tower landing zone with Landing Zone Accelerator (LZA) on AWS .

LZA is a solution that deploys a foundational set of capabilities designed to align with AWS best practices, and with multiple global compliance frameworks, to help you manage and govern a multi-account environment. LZA is built using the AWS Cloud Development Kit (CDK).

LZA automatically sets up a cloud environment that's suitable for hosting secure workloads. This solution can be deployed in all AWS Regions, to help you maintain consistency of operations and governance. The LZA solution is architected to align with AWS best practices, and to conform with multiple global compliance frameworks.

We recommend that you deploy AWS Control Tower as your foundational landing zone, and then enhance your landing zone capabilities with LZA, as needed. The combination of LZA and AWS Control Tower provides a comprehensive, no-code solution that helps you manage and govern a multi-account environment—it's built to support highly-regulated workloads and complex compliance requirements. Together, AWS Control Tower and the Landing Zone Accelerator help you establish platform readiness, including security, compliance, and operational capabilities.

Source code for LZA is available in a GitHub repository.

For more information on how to combine LZA and AWS Control Tower, see the LZA implementation guide.

Benefits of Customizations for AWS Control Tower (CfCT)

The package of functionality that we refer to as *Customizations for AWS Control Tower* (CfCT) helps you create more extensive customizations for your landing zone than you can create in the AWS Control Tower console. It offers a GitOps-style, automated process. You can reshape your landing zone to meet your business requirements.

This *infrastructure-as-code* customization process integrates AWS CloudFormation templates with AWS service control policies (SCPs) and AWS Control Tower lifecycle events, so that your resource deployments remain synchronized with your landing zone. For example, when you create a new account with Account Factory, the resources attached to the account and the OU can be deployed automatically.

> **Note**
>
> Unlike Account Factory and AFT, CfCT is not specifically intended to create new accounts, but to customize accounts and OUs in your landing zone by deploying resources that you specify.

Benefits

- **Expand a customized and secure AWS environment –** You can expand your multi-account AWS Control Tower environment more quickly, and incorporate AWS best practices into a repeatable customization workflow.

- **Instantiate your requirements –** You can customize your AWS Control Tower landing zone for your business requirements, with the AWS CloudFormation templates and service control policies that express your policy intentions.
- **Automate further with AWS Control Tower lifecycle events –** Lifecycle events allow you to deploy resources based on completion of a previous series of events. You can rely on a lifecycle event to help you deploy resources to accounts and OUs, automatically.
- **Extend your network architecture –** You can deploy customized network architectures that improve and protect your connectivity, such as a transit gateway.

Additional CfCT examples

- An example networking use case with *Customizations for AWS Control Tower* (CfCT) is given in the AWS Architecture blog post, Deploy consistent DNS with Service Catalog and AWS Control Tower customizations.
- A specific example related to CfCT and Amazon GuardDuty is available on GitHub in the `aws-samples` repository.
- Additional code examples regarding CfCT are available as part of the AWS Security Reference Architecture, in the `aws-samples` repository. Many of these examples contain sample `manifest.yaml` files in a directory named `customizations_for_aws_control_tower`.

For more information about the AWS Security Reference Architecture, see the AWS Prescriptive Guidance pages.

Customizations for AWS Control Tower (CfCT) overview

Customizations for AWS Control Tower (CfCT) helps you customize your AWS Control Tower landing zone and stay aligned with AWS best practices. Customizations are implemented with AWS CloudFormation templates and service control policies (SCPs).

This CfCT capability is integrated with AWS Control Tower lifecycle events, so that your resource deployments remain synchronized with your landing zone. For example, when a new account is created through account factory, all resources attached to the account are deployed automatically. You can deploy the custom templates and policies to individual accounts and organizational units (OUs) within your organization.

The following video describes best practices for deploying a scalable CfCT pipeline and common CfCT customizations.

The following section provides architectural considerations and configuration steps for deploying Customizations for AWS Control Tower (CfCT). It includes a link to the AWS CloudFormation template that launches, configures, and runs the required AWS services, in alignment with AWS best practices for security and availability.

This topic is intended for IT infrastructure architects and developers who have practical experience architecting in the AWS Cloud.

For information about the latest updates and changes to Customizations for AWS Control Tower (CfCT), refer to the CHANGELOG.md file in the GitHub repository.

Architecture overview

Deploying CfCT builds the following environment in the AWS Cloud, with an Amazon S3 bucket as a configuration source.

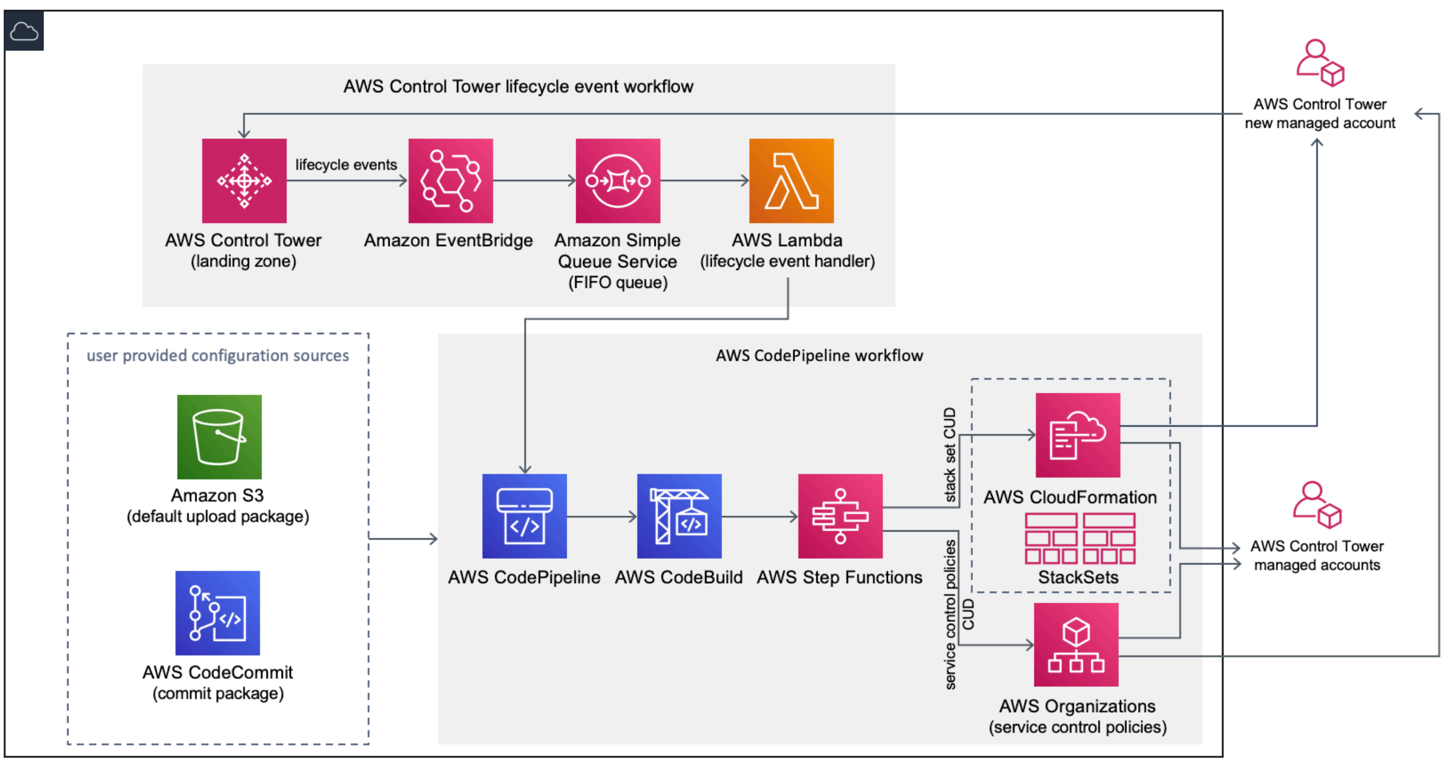

Figure 1: Customizations for AWS Control Tower architecture

CfCT includes an AWS CloudFormation template that you deploy in your AWS Control Tower management account. The template launches all the components necessary to build the workflows, so you can customize your AWS Control Tower landing zone.

Note

CfCT must be deployed in the AWS Control Tower home Region and in the AWS Control Tower management account, because that is where your AWS Control Tower landing zone is deployed. For information about setting up an AWS Control Tower landing zone, refer to *Getting started*.

As you deploy CfCT, it packages and uploads the custom resources to the code pipeline source, by means of Amazon Simple Storage Service (Amazon S3). The upload process automatically invokes the service control policies (SCPs) state machine and the AWS CloudFormation StackSets state machine to deploy the SCPs at the OU level, or to deploy stack instances at the OU or account level.

Note

By default, CfCT creates an Amazon S3 bucket to store the pipeline source. If you have an existing AWS CodeCommit repository, you can change the location to a CodeCommit repository. For more information, refer to Set up Amazon S3 as the configuration source.

CfCT deploys two workflows:

- an AWS CodePipeline workflow
- and an AWS Control Tower lifecycle event workflow.

The AWS CodePipeline workflow

The AWS CodePipeline workflow configures AWS CodePipeline, AWS CodeBuild projects, and AWS Step Functions that orchestrate the management of AWS CloudFormation StackSets and SCPs in your organization.

When you upload the configuration package, CfCT invokes the code pipeline to run three stages.

- **Build Stage** – validates the contents of the configuration package using AWS CodeBuild.

- **SCP Stage –** invokes the service control policy state machine, which calls the AWS Organizations API to create SCPs.
- **AWS CloudFormation Stage –** invokes the stack set state machine to deploy the resources specified in the list of accounts or OUs, which you've provided in the manifest file.

At each stage, the code pipeline invokes the stack set and SCP step functions, which deploy custom stack sets and SCPs to the targeted individual accounts, or to an entire organizational unit.

> **Note**
>
> For detailed information about customizing the configuration package, refer to CfCT customization guide.

The AWS Control Tower lifecycle event workflow

When a new account is created in AWS Control Tower, a lifecycle event can invoke the AWS CodePipeline workflow. You can customize the configuration package through this workflow, which consists of an Amazon EventBridge event rule, an Amazon Simple Queue Service (Amazon SQS) first-in first-out (FIFO) queue, and an AWS Lambda function.

When the Amazon EventBridge event rule detects a matching lifecycle event, it passes the event to the Amazon SQS FIFO queue, invokes the AWS Lambda function, and invokes the code pipeline to perform downstream deployment of stack sets and SCPs.

Cost

The cost for running CfCT depends on the number of AWS CodePipeline runs, the duration of AWS CodeBuild runs, the number and duration of AWS Lambda functions, and the number of Amazon EventBridge events published. For example, if you run 100 builds in one month using **build.general1.small** where each build runs for five minutes, then the approximate cost for running CfCT is **$3.00 per month**. For full details, you can review the pricing webpage for each AWS service you are running.

The Amazon Simple Storage Service (Amazon S3) bucket and AWS CodeCommit Git-based repository resources are retained after you delete the template, to protect your configuration information. Depending on the option you select, you are charged based on the amount of data stored in the Amazon S3 bucket and the number of Git requests (not applicable to Amazon S3 resource). Refer to Amazon S3 and AWS CodeCommit pricing for details.

Component services

The following AWS services are components of *Customizations for AWS Control Tower* (CfCT).

AWS CodeCommit

If you have an existing AWS CodeCommit repository, you can configure it as a source for your pipeline, as an alternative to Amazon S3.

Based on your input to the AWS CloudFormation template, CfCT can create an AWS CodeCommit repository with the same sample configuration that's explained in the Amazon Simple Storage Service section.

To clone the CfCT AWS CodeCommit repository to your local computer, you must create credentials that give you temporary access to the repository, as explained in the AWS CodeCommit User Guide. For information about version compatibility, see Setting up for AWS CodeCommit.

> **Note**
>
> If you do not already use CodeCommit, your only option is to set up the Amazon S3 bucket as the storage location for your configuration package. CodeCommit is not available if you are deploying CfCT for the first time.

AWS CodePipeline

AWS CodePipeline validates, tests, and implements changes based on updates to the configuration package, which you'll make in either the default Amazon S3 bucket or the AWS CodeCommit repository. For more information about configuration source control, see Using Amazon S3 as the Configuration Source. The pipeline includes stages to validate and manage the configuration files and templates, core accounts, AWS Organizations service control policies, and AWS CloudFormation StackSets. For more information about the pipeline stages, refer to CfCT customization guide

AWS Key Management Service

CfCT creates an AWS Key Management Service (AWS KMS) `CustomControlTowerKMSKey` encryption key. This key is used to encrypt objects in the Amazon S3 configuration bucket, Amazon SQS queue, and sensitive parameters in the AWS Systems Manager Parameter Store. By default, only roles provisioned by CfCT have permission to perform encryption or decryption operations with this key. For access to the configuration file, FIFO queue, or Parameter Store `SecureString` values, administrators must be added to the `CustomControlTowerKMSKey` policy. Automatic key rotation is enabled by default.

AWS Lambda

CfCT uses AWS Lambda functions to invoke the installation components during the initial installation and deployment of AWS CloudFormation StackSets or AWS Organizations SCPs during an AWS Control Tower lifecycle event.

Amazon Simple Notification Service

CfCT may publish notifications, such as pipeline approval to Amazon Simple Notification Service (Amazon SNS) topics during the workflow. Amazon SNS is launched only when you choose to receive pipeline approval notifications.

Amazon Simple Storage Service

When you deploy CfCT, CfCT creates an Amazon Simple Storage Service (Amazon S3) bucket with a unique name:

Example: Amazon S3 bucket name

```
custom-control-tower-configuration-accountID-region
```

The bucket contains a sample configuration file called `_custom-control-tower-configuration.zip`

Notice the leading underscore in the file name.

This zip file provides a sample manifest and the related sample templates that describe the necessary folder structure. These examples help you develop a configuration package to customize your AWS Control Tower landing zone. The sample manifest identifies the required configurations for stack sets and service control policies (SCPs) you'll need, when you implement your customizations.

You can use this sample configuration package as a model, to develop and upload your custom package, which triggers the CfCT configuration pipeline automatically.

For information about customizing the configuration file, see CfCT customization guide.

Amazon Simple Queue Service

CfCT uses an Amazon Simple Queue Service (Amazon SQS) FIFO queue to capture lifecycle events from Amazon EventBridge. It triggers an AWS Lambda function, which invokes AWS CodePipeline to deploy AWS CloudFormation StackSets or SCPs. For more information about SCPs, see AWS Organizations.

AWS Step Functions

CfCT creates Step Functions to orchestrate customization deployments. These Step Functions translate configuration files to deploy the customizations as needed across environments.

AWS Systems Manager Parameter Store

AWS Systems Manager Parameter Store stores the CfCT configuration parameters. These parameters allow you to integrate related configuration templates. For example, you can configure each account to log AWS CloudTrail data to a centralized Amazon S3 bucket. Also, the Systems Manager Parameter Store provides a centralized location where administrators can view CfCT inputs and parameters.

Deployment considerations

Be sure to launch *Customizations for AWS Control Tower* (CfCT) in the same account and Region where your AWS Control Tower landing zone is deployed; that is, you must deploy it in the AWS Control Tower management account in your AWS Control Tower home Region. By default, CfCT creates and runs the landing zone configuration package by setting up a configuration pipeline in that account and Region.

Prepare for deployment

You have some options when you prepare your AWS CloudFormation template for initial deployment. You can choose the configuration source, and you can allow for manual approval of pipeline deployments. The next two sections explain more about these options.

Choose your configuration source

By default, the template creates an Amazon Simple Storage Service (Amazon S3) bucket to store the sample configuration package as a `.zip` file called `_custom-control-tower-configuration.zip`. The Amazon S3 bucket is version controlled, and you can update the configuration package as needed. For information about updating the configuration package, refer to Using Amazon S3 as the Configuration Source.

> **Remember to remove the underscore**
>
> The sample configuration package filename begins with an underscore (_) so that AWS CodePipeline is not initiated automatically. When you have finished customizing the configuration package, be sure to upload the `custom-control-tower-configuration.zip` without the underscore (_) in order to begin the deployment in AWS CodePipeline.

If you have an existing AWS CodeCommit Git repository, you can change the storage location of the configuration package from the Amazon S3 bucket to an AWS CodeCommit Git repository. To do so, select the `CodeCommit` option in the AWS CloudFormation parameter.

> **To zip, or not to zip?**
>
> When you're using the default S3 bucket, be sure that the configuration package is available as a `.zip` file. If you're using the AWS CodeCommit repository, be sure to place the configuration package in the repository without zipping the files. For information about creating and storing the configuration package in AWS CodeCommit, see CfCT customization guide.

You can use the sample configuration package to create your own custom configuration source. When you are ready to deploy your custom configurations, manually upload the configuration package, either to the Amazon S3 bucket or to the AWS CodeCommit repository. The pipeline begins automatically when you upload the configuration file.

Choose your pipeline configuration approval parameters

The AWS CloudFormation template provides the option to approve the deployment of configuration changes manually. By default, manual approval is not enabled. For more information, refer to Step 1. Launch the stack.

When manual approval is enabled, the configuration pipeline validates the customizations made to the AWS Control Tower file manifest and templates, then it pauses the process until manual approval is granted. After approval, the deployment proceeds to run the remaining pipeline stages, as needed, to implement the *Customizations for AWS Control Tower* (CfCT) functionality.

You can use the manual approval parameter to keep the customizations for the AWS Control Tower configuration from running, by rejecting the first attempt to run through the pipeline. This parameter also allows you to validate customizations for the AWS Control Tower configuration changes manually, as a final control before implementation.

To update Customizations for AWS Control Tower

If you have previously deployed CfCT, you must update the AWS CloudFormation stack to get the latest version of the CfCT framework. For details, refer to Update the Stack.

Template and source code

Customizations for AWS Control Tower (CfCT) are deployed in your management account after you launch your AWS CloudFormation template. You can download the template from GitHub and then launch it from AWS CloudFormation.

The **customizations-for-aws-control-tower.template** deploys the following:

- An AWS CodeBuild project
- An AWS CodePipeline project
- An Amazon EventBridge rule
- AWS Lambda functions
- An Amazon Simple Queue Service queue
- An Amazon Simple Storage Service bucket with a sample configuration package
- AWS Step Functions

> **Note**
>
> You can customize the template based on your specific requirements.

Source code repository

You can visit our GitHub repository to download the templates and scripts for CfCT, and to share your landing zone customizations with others.

Automated deployment

Before you launch the automated deployment, review the considerations. Follow the step-by-step instructions in this section to configure and deploy the solution into your AWS Control Tower management account.

Time to deploy: Approximately 15 minutes

Prerequisites

CfCT must be deployed in your AWS Control Tower management account, and in your AWS Control Tower home Region. If you do not have a landing zone set up, see *Getting started*.

Deployment steps

The procedure for deploying CfCT consists of two major steps. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your management account.
- Review the template parameters, and adjust if necessary.

Step 2. Create a custom package

- Create a custom configuration package.

> **Important**
>
> To download the correct AWS CloudFormation template and launch CfCT, follow the GitHub link given in this section. Do not follow older links to any previously specified S3 buckets.

Step 1. Launch the stack

The AWS CloudFormation template in this section deploys *Customizations for AWS Control Tower* (CfCT) in your account.

> **Note**
>
> You are responsible for the cost of the AWS services used while you run CfCT. For more details, see Cost.

1. To launch *Customizations for AWS Control Tower*, download the template from GitHub and then launch it from AWS CloudFormation.
2. The template launches in the US East (N. Virginia) Region by default. To launch CfCT in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > CfCT must be launched in the same Region and account where you deployed your AWS Control Tower landing zone, which is your home Region.
3. On the **Create stack** page, verify that the correct template URL shows in the **URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your CfCT stack.
5. Under **Parameters**, review the following parameters and modify them in the template, if necessary.

Pipeline Configuration		
Parameter	**Default**	**Description**
Pipeline Approval Stage	No	Choose whether to change the pipeline configuration from the default automated approval stage to a manual approval stage. For more information, see the section

Pipeline Configuration		
Parameter	**Default**	**Description**
		called "CfCT customization guide".
Pipeline Approval Email Address	<Optional Input>	The email address for approval notifications. To use this parameter, you must set the **Pipeline Approval Stage** parameter to `Yes`.
AWS CodePipeline Source	`Amazon S3`	The source for AWS CodePipeline to help you select where to store and configure the CfCT customizations.

AWS CodeCommit Setup		
Parameter	**Default**	**Description**
Existing CodeCommit Repository?	No	Choose whether to use an existing CodeCommit Git repository. If you choose `Yes`, you must set the **CodePipeline Source** parameter to `AWS CodeCommit` .
CodeCommit Repository Name	`custom-control-tower-configuration`	If you provide the name of an existing Git repository, you must set the **Existing CodeCommit Repository?** parameter to `Yes` and enter the exact name of that repository.

AWS CodeCommit Setup		
Parameter	**Default**	**Description**
CodeCommit Branch Name	`main`	The Git branch where the customization package is stored. To use this parameter, you must set the **CodePipeline Source** parameter to `AWS CodeCommit` .
AWS CloudFormation StackSets Configuration		
Parameter	**Default**	**Description**
Region Concurrency Type	`PARALLEL`	Select the concurrency type of deploying StackSets operations in Regions. This setting is applicable for create, update, and delete workflows. Other allowed value is `SEQUENTIAL` .
Max Concurrent Percentage	`100`	The maximum percentage of accounts in which to perform this operation at one time. The max allowed value is 100. For more information, refer to Stack Set operation options.

AWS CloudFormation StackSets Configuration		
Parameter	**Default**	**Description**
Failure Tolerance Percentage	`10`	The percentage of accounts, per Region, for which this stack operation can fail before AWS CloudFormation stops the operation in that Region. The minimum allowed value is 0 and max allowed value is 100. For more information, refer to [Stack Set operation options]().

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately 15 minutes.

Step 2. Create a custom package

With the launched stack, you can add customizations to your AWS Control Tower landing zone and service control policies (SCPs) by customizing the included configuration package. For detailed instructions on creating a custom package, refer to the [CfCT customization guide]().

> **Note**
>
> The pipeline does not run without uploading the custom configuration package.

Update the stack

If you previously deployed *Customizations for AWS Control Tower* (CfCT), follow the procedure to update the AWS CloudFormation stack for the latest version of the CfCT framework.

> **Important**
>
> Before you can complete the following procedure, you must upload the latest template from GitHub to an Amazon Simple Storage Service (Amazon S3) bucket. For instructions on how to get started with Amazon S3, see Getting started with Amazon S3 in the *Amazon Simple Storage Service User Guide*.

1. Sign in to the AWS CloudFormation console.
2. Select your existing **Customizations for AWS Control Tower** (CfCT) CloudFormation stack, and then select **Update**.
3. Under **Prerequisite — Prepare template**, select **Replace current template**.
4. Under **Specify template**, do the following:

    a. For **Template source**, select **Replace current template**.

    b. For **Amazon S3 URL**, enter the template URL for the template that you previously uploaded from GitHub to Amazon S3, and then choose **Next**.

    c. Verify that the template URL is correct. Then choose **Next** and **Next** again.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the stack for details about the parameters.
6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create AWS Identity and Access Management (IAM) resources.
9. Choose **View change set** and verify the changes.
10. Choose **Update stack** to deploy the stack.

    You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **UPDATE_COMPLETE** in approximately 15 minutes.

Delete a stack set

You can delete a stack set if you've enabled stack set deletion in the manifest file. By default, the `enable_stack_set_deletion` parameter is set to `false`. In this configuration, no action is taken to delete the associated stack set when a resource is removed from the CfCT manifest file.

If you change the value of `enable_stack_set_deletion` to `true` in the manifest file, CfCT deletes the stack set and all of its resources when you remove an associated resource from the manifest file.

This capability is supported in **v2** of the manifest file.

> **Important**
>
> When you initially set the value of `enable_stack_set_deletion` to `true`, the next time you invoke CfCT, **ALL** resources that begin with the prefix `CustomControlTower-`, which have the associated key tag `Key:AWS_Solutions, Value: CustomControlTowerStackSet`, and which are not declared in the manifest file, are staged for deletion.

Here's an example of how to set this parameter in a `manifest.yaml` file:

```
version: 2021-03-15
region: us-east-1
enable_stack_set_deletion: true   #New opt-in functionality


resources:
  - name: demo_resource_1
    resource_file: s3://demo_bucket/resource.template
    deployment_targets:
      accounts:
        - 012345678912
    deploy_method: stack_set
    ...
    regions:
    - us-east-1
    - us-west-2

  - name: demo_resource_2
```

```
    resource_file: s3://demo_bucket/resource.template
    deployment_targets:
      accounts:
        - 012345678912
    deploy_method: stack_set
    ...
    regions:
    - us-east-1
    - eu-north-1
```

Set up Amazon S3 as the configuration source

When you set up *Customizations for AWS Control Tower*, it stores an initial configuration file, called `_custom-control-tower-configuration.zip` file in an Amazon Simple Storage Service (Amazon S3) bucket, named `custom-control-tower-configuration-`*`account-ID`*`-`*`region`*.

> **Note**
>
> If you choose to download and modify this file, remember to zip the changes, save as a new file named `custom-control-tower-configuration.zip`, and then upload it back to the same Amazon S3 bucket.
> The Amazon S3 bucket is the default source of the pipeline. When default settings are in place, uploading a configuration zip file without the underscore prefix in the file name to the S3 bucket will initiate the pipeline automatically.

The zip file is protected by Server-Side Encryption (SSE) with AWS Key Management Service (AWS KMS), and denial of use of the KMS key. For access to the zip file, you must update the KMS Key Policy to specify the role(s) that should be granted access. The role may be an administrator role, a user, or both. Follow this procedure:

1. Navigate to the AWS Key Management Service console.
2. In **Customer Managed Keys**, select **CustomControlTowerKMSKey**.
3. Select the **Key policy** tab. Then, select **Edit**.
4. In the **Edit key policy** page, find the **Allow Use of the key** section in the code, and add one of the following permissions:

- To add an administration role:

  `arn:aws:iam::<account-ID>:role/<administrator-role>`

- To add a user::

  `arn:aws:iam::<account-ID>:user/<username>`

5. Select **Save Changes**.
6. Navigate to the Amazon S3 console, find the S3 bucket containing the configuration zip file, and select download.
7. Make the necessary configuration changes to the manifest file and template files. For information about customizing the manifest and template files, see the section called “CfCT customization guide”.
8. Upload your changes:
   a. Zip the modified configuration files, and name the file: `custom-control-tower-configuration.zip`.
   b. Upload the file to Amazon S3 using SSE with the AWS KMS master-key: `CustomControlTowerKMSKey`.

Set up GitHub as the configuration source

This section tells you how to deploy Customizations for AWS Control Tower (CfCT) with GitHub as a source. The process has three main steps:

- Prepare a GitHub repository
- Create the GitHub code connection
- Deploy the AWS CloudFormation stack

Prepare a GitHub repository

Create a repository within your GitHub account, the default name used in the template is `custom-control-tower-configuration`. Consider making the target repository *private*. You'll define your customizations in a `yaml` file called `manifest.yaml` in the deployment folder of the CfCT repository.

The *CfCT customization guide* provides detailed guidance on creating a `manifest.yaml` to configure your customizations.

Create the GitHub conection

From your **Developer Tools --Connections** instance for Github, perform the following steps:

1. Select **Create connection** and choose GitHub as the provider
2. Choose **Create a GitHub App connection**, and in the **Connection name** field, enter **GitHub CfCT**, or any name you choose
3. Select **Connect to GitHub** and then choose **Install a new app**
4. Select the GitHub User or Organization for your repository
5. Under **Repository access**, choose **Only select repositories**, then select the repository you created earlier, and **Save** your work.
6. Note the Code Connections ARN - you'll need it when deploying the AWS CloudFormation stack.

Deploy the AWS CloudFormation stack

- Download the `custom-control-tower-initiation.template` file from the repository.
- Create a new AWS CloudFormation stack, using the `custom-control-tower-initiation.template` file.
- Under **AWS CodePipeline Source**, choose **GitHub (via Code Connection)**.
- Under **GitHub Setup**, specify these fields:
  - For **ARN of the Code Connection**, provide the Code Connection ARN
  - for **GitHub User or Organization**, provide the name of the GitHub user or organization under which you created the repository
  - For **GitHub Repository Name**, enter the repository name (defaults to `custom-control-tower-configuration`)
  - For **GitHub Branch Name**, enter the branch name (defaults to `main`)

Collection of operational metrics

Customizations for AWS Control Tower (CfCT) includes an option to send anonymous operational metrics to AWS. AWS uses this data to understand how customers are using CfCT, as well as other related services and products. When data collection is enabled, the following information is sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each deployment
- **Timestamp:** Data-collection timestamp
- **State Machine Execution Count:** Incrementally counts the number of times this state machine runs
- **Manifest Version:** The manifest version used in the configuration

> **Note**
>
> AWS owns the data it collects. Data collection is subject to the AWS Privacy Policy.

To opt out of sending anonymous operational metrics to AWS, complete one of the following tasks:

- **Update the AWS CloudFormation template mapping section as follows:**

  from

  ```
  AnonymousData:
       SendAnonymousData:
          Data: Yes
  ```

  to

  ```
  AnonymousData:
       SendAnonymousData:
          Data: No
  ```

- **After CfCT is deployed, find the `/org/primary/metrics_flag` SSM parameter key in the Parameter Store console, and update the value to No.**

CfCT customization guide

The *Customizations for AWS Control Tower* (CfCT) guide is for administrators, DevOps professionals, independent software vendors, IT infrastructure architects, and systems integrators who want to customize and extend their AWS Control Tower environments for their company and customers. It provides information about customizing and extending the AWS Control Tower environment with the CfCT customization package.

> **Note**
>
> To deploy and configure (CfCT), you must deploy and process a configuration package through AWS CodePipeline. The following sections describe the process in detail.

Code pipeline overview

The configuration package requires Amazon Simple Storage Service (Amazon S3) and AWS CodePipeline. The configuration package contains these items:

- A manifest file
- An accompanying set of templates
- Other JSON files for describing and implementing your AWS Control Tower environment customizations

By default, the `_custom-control-tower-configuration.zip` configuration package is loaded in an Amazon S3 bucket with the following naming convention:

`custom-control-tower-configuration-*accountID*-*region*`.

> **Note**
>
> By default, CfCT creates an Amazon S3 bucket to store the pipeline source. Most customers stay with this default. If you have an existing AWS CodeCommit repository, you can change the source location to your AWS CodeCommit repository. For more information, see [Edit a pipeline in CodePipeline]() in the *AWS CodePipeline User Guide*.

The *manifest file* is a text file that describes the AWS resources you can deploy to customize your landing zone. CodePipeline does these tasks:

- extracts the manifest file, accompanying set of templates, and other JSON files
- performs manifest and template validations
- invokes sections in the CfCT manifest file to run specific pipeline stages.

When you update the configuration package by customizing the manifest file and removing the underscore (_) from the configuration package filename, it automatically initiates AWS CodePipeline.

Remember the underscore

The sample configuration package filename begins with an underscore (_) so that AWS CodePipeline is not automatically triggered. When you have completed the customization of the configuration package, upload the file `custom-control-tower-configuration.zip` without the underscore (_) in order to trigger the deployment in AWS CodePipeline.

AWS CodePipeline stages

The CfCT pipeline requires several AWS CodePipeline stages to implement and update your AWS Control Tower environment.

1. **Source stage**

   The source stage is the initial stage. Your customized configuration package initiates this pipeline stage. The source for the AWS CodePipeline can be either an Amazon S3 bucket or an AWS CodeCommit repository, in which the configuration package can be hosted.

2. **Build stage**

   The build stage requires AWS CodeBuild to validate the contents of the configuration package. These checks include testing the `manifest.yaml` file syntax and schema, along with all AWS CloudFormation templates included in the package or remotely hosted, using AWS CloudFormation `validate-template` and `cfn_nag`. If the manifest file and AWS CloudFormation templates pass the tests, the pipeline continues to the next stage. If the tests fail, you can review the CodeBuild logs to identify the issue and edit the configuration source file as needed.

3. **Manual approval stage (optional)**

   The manual approval stage is optional. If you enable this stage, it provides additional control over the configuration pipeline. It pauses the pipeline during deployment, until an approval is given. You can opt into manual approval by editing the **Pipeline Approval Stage** parameter to **Yes** when you launch the stack.

4. **Policy stage**

   The policy stage invokes the service control policy (SCP) or resource control policy (RCP) state machine to call AWS Organizations APIs that create SCPs or RCPs.

5. **AWS CloudFormation resource stage**

   The AWS CloudFormation resource stage invokes the stack set state machine to deploy the resources specified in the list of accounts or organizational units (OUs), which you provided in the manifest file. The state machine creates the AWS CloudFormation resources in the order that they are specified in the manifest file. To specify a resource dependency, arrange the order in which resources are specified in the manifest file. The order of resources within the manifest file is the only way to specify a dependency.

Define a custom configuration

You'll define your custom AWS Control Tower configuration with the CfCT manifest file, the accompanying set of templates, and other JSON files. You'll package these files into a folder structure and place them in the Amazon S3 bucket as a `.zip` file, as shown in the following code example.

Custom configuration folder structure

```
- manifest.yaml
- policies/                                                [optional]
   - service control policies files (*.json)
- templates/                                               [optional]
   - template files for AWS CloudFormation Resources (*.template)
```

The previous example depicts the structure of a custom configuration folder. The folder structure stays the same whether you choose Amazon S3 or an AWS CodeCommit repository as your source storage location. If you choose Amazon S3 as source storage, compress all the folders and files into a `custom-control-tower-configuration.zip` file, and upload only the `.zip` file to the designated Amazon S3 bucket.

> **Note**
>
> If you are using AWS CodeCommit, place the files in the repository without zipping the files.

The CfCT manifest file

The CfCT `manifest.yaml` file is a text file that describes your AWS resources. The following example shows the structure of the CfCT manifest file.

```
---
region: String
version: 2021-03-15

resources:
  #set of CloudFormation resources, SCP policies, or RCP policies
...
```

As shown in the previous code example, the first two lines of the manifest file specify the values of the **region** and the **version** keywords. Here are the definitions of those keywords.

region – A text string for the AWS Control Tower default Region. This value must be a valid AWS Region name (such as `us-east-1`, `eu-west-1`, or `ap-southeast-1`). The AWS Control Tower home Region is the default when you create custom AWS Control Tower resources (such as AWS CloudFormation StackSets), unless a more resource-specific Region is specified.

```
region:your-home-region
```

version – The manifest schema version number. The latest supported version is 2021-03-15.

```
version: 2021-03-15
```

> **Note**
>
> We strongly recommend you use the latest version. To update manifest properties in the latest version, refer to Version upgrades for the CfCT manifest.

The next keyword shown in the previous example is the **resources** keyword. The **resources** section of the manifest file is highly structured. It contains a detailed list of AWS resources, which will be deployed automatically by the CfCT pipeline. These descriptions of resources and their available parameters are given in the next section.

The resources section of the CfCT manifest file

This topic describes the **resources** section of the CfCT manifest file, where you'll define the resources that are required for your customizations. This section of the CfCT manifest file begins at the keyword **resources** and continues to the end of the file.

The **resources** section of the manifest file specifies the AWS CloudFormation StackSets, or AWS Organizations SCPs and RCPs, which CfCT deploys automatically through the code pipeline. You can list OUs, accounts, and Regions to deploy stack instances.

Stack instances are deployed at the account level instead of the OU level. SCPs and RCPs are deployed at the OU level. For more information, see Build your own customizations.

The following example template describes the possible entries that are available for the **resources** section of the manifest file.

```
resources: # List of resources
  - name: [String]
    resource_file: [String] [Local File Path, S3 URI, S3 URL]
    deployment_targets: # account and/or organizational unit names
      accounts: # array of strings, [0-9]{12}
        - 012345678912
        - AccountName1
      organizational_units: #array of strings
        - OuName1
        - OuName2
    deploy_method: scp | stack_set | rcp
    parameters: # List of parameters [SSM, Alfred, Values]
      - parameter_key: [String]
```

```
        parameter_value: [String]
    export_outputs: # list of ssm parameters to store output values
      - name: /org/member/test-ssm/app-id
        value: $[output_ApplicationId]
    regions: #list of strings
    - [String]
```

The remainder of this topic gives detailed definitions for the keywords shown in the previous code example.

name – The name that is associated with the AWS CloudFormation StackSets.
The string you provide assigns a more user-friendly name for a stack set.

- **Type:** String
- **Required:** Yes
- **Valid Values:** a-z, A-Z, 0-9, and an underscore (_). Any other character is automatically replaced with an underscore (_).

description – The description for the resource.

- **Type:** String
- **Required:** No

resource_file – This file can be specified as the relative location to the manifest file, an Amazon S3 URI or URL that points to an AWS CloudFormation template or AWS Organizations service control policy in JSON for creating AWS CloudFormation resources, SCPs, or RCPs.

- **Type:** String
- **Required:** Yes

1. The following example shows the `resource_file`, given as a relative location to the resource file inside the configuration package.

   ```
   resources:
     - name: SecurityRoles
       resource_file: templates/custom-security.template
   ```

2. The following example shows the resource file given as an Amazon S3 URI

```
resources:
  - name: SecurityRoles
    resource_file: s3://amzn-s3-demo-bucket/[key-name]
```

3. The following example shows the resource file given as an Amazon S3 HTTPS URL

```
resources:
  - name: SecurityRoles
    resource_file: https://bucket-name.s3.Region.amazonaws.com/key-name
```

> **Note**
>
> If you provide an Amazon S3 URL, verify that the bucket policy allows read access for the AWS Control Tower management account from which you are deploying CfCT. If you provide an Amazon S3 HTTPS URL, verify that the path uses dot notation. For example, `S3.us-west-1`. CfCT does not support endpoints that contain a dash between S3 and the Region, such as `S3-us-west-2`.

4. The following example shows an Amazon S3 bucket policy and an ARN where resources are stored.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::AccountId:root"},
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::my-bucket/*”
       }
   ]
}
```

You'll replace the *`AccountId`* variable shown in the example with the AWS account ID for the management account that is deploying CfCT. For more examples, refer to [Bucket policy examples](#) in the Amazon Simple Storage Service User Guide.

parameters – Specifies the name and value for AWS CloudFormation parameters.

- **Type:** MapList
- **Required:** No

The parameters section contains pairs of key/value parameters. The following pseudo template outlines the **parameters** section.

```
parameters:
  - parameter_key: [String]
    parameter_value: [String]
```

- **parameter_key** – The key associated with the parameter.
  - **Type:** String
  - **Required:** Yes (under parameters property)
  - **Valid Values:** a-z, A-Z, and 0-9
- **parameter_value** – The input value associated with the parameter.
  - **Type:** String
  - **Required:** Yes (under parameters property)

deploy_method – The deployment method for deploying resource(s) into the account. Currently, **deploy_method** supports deploying resources using the `stack_set` option for resource deployment through AWS CloudFormation StackSets, the `scp` option if you are deploying SCPs, or the `rcp` option if you are deploying RCPs.

- **Type:** String
- **Valid Values:** `stack_set` | `scp` | `rcp`
- **Required:** Yes

deployment_targets – List of accounts or Organizational Units (OUs), into which CfCT will deploy the AWS CloudFormation resources, specified as **accounts** or **organizational_units**.

> **Note**
>
> If you want to deploy an SCP or RCP, the target must be an OU, not an account.

- **Type:** List of string `account name` or `account number` to indicate that this resource will be deployed into the given account list, or `OU names` to indicate that this resource will be deployed into the given OU list.
- **Required:** At least one of **accounts** or **organizational_units**
  - **accounts:**

    Type: List of string `account name` or `account number` to indicate that this resource will be deployed into the given account list.
  - **organizational_units:**

    Type: List of string `OU names` to indicate that this resource will be deployed into a given OU list. If you provide an OU that doesn't contain accounts and the **accounts** property is not added, CfCT only creates the stack set.

> **Note**
>
> The organization's management account ID is not an allowed value. CfCT does not support deploying stack instances into the organization's management account, by default. If you have a special use case, see Root OU.

export_outputs – List of name/value pairs that denote SSM parameter keys. These SSM parameter keys allow you to store template outputs into the SSM parameter store. The output is intended for reference by other resources, defined earlier in the manifest file.

```
export_outputs: # List of SSM parameters
  - name: [String]
    value: [String]
```

- **Type:** List of **name** and **value** key pairs. The **name** contains the `name` string of an SSM parameter store key, and **value** contains the parameter's `value` string.
- **Valid Values:** Any string or the `$[output_CfnOutput-Logical-ID]` variable where *`CfnOutput-Logical-ID`* corresponds to the template output variable. For more information about the Outputs section in an AWS CloudFormation template, see **Outputs** in the *AWS CloudFormation User Guide*.
- **Required:** No

For example, the following code snippet stores the template VPCID output variable into the SSM parameter key that's named /org/member/audit/vpc_id.

```
export_outputs: # List of SSM parameters
  - name: /org/member/audit/VPC-ID
    value: $[output_VPCID]
```

> **Note**
>
> The **export_outputs** key name may contain a value other than output. For example, if the **name** is /org/environment-name, the **value** may be production.

regions – List of Regions in which CfCT will deploy the AWS CloudFormation stack instances.

- **Type:** Any list of AWS commercial Region names, to indicate that this resource will be deployed into the given Region list. If this keyword does not exist in the manifest file, the resources are deployed in the home Region only.
- **Required:** No

Root OU

CfCT supports **Root** as a value for an organizationl unit (OU) under organizational_units in **manifest V2 version (2021-03-15)**.

- If you choose the deployment method of scp or rcp, when you add Root under organizational_units, AWS Control Tower applies the policies to all of the OUs under the Root. If you choose the deployment method of stack_set, when you add Root under organizational_units, CfCT deploys the stack sets in all the accounts under the Root that are enrolled in AWS Control Tower, except for the management account.
- As per AWS Control Tower best practices, the management account is intended only to manage member accounts and for billing purposes. Do not run production workloads in the AWS Control Tower management account.

  In accordance with best practices guidance, AWS Control Tower deployment puts the management account under the Root OU, so that it has full access and does not run additional resources. For this reason, the **AWSControlTowerExecution** role is not deployed to the management account.

- We recommend that you follow these best practices for the management account. If you have a specific use case that requires you to deploy stacksets in the management account, include **accounts** as a deployment target and specify the management account. Otherwise, do not include **accounts** as a deployment target. You must create the missing resources, including required IAM roles, in the management account.

To deploy stacksets in the management account, include `accounts` as a deployment target and specify the management account. Otherwise, do not include accounts as a deployment target.

```
---
region: your-home-region
version: 2021-03-15

resources:

  …truncated…

    deployment_targets:
      organizational_units:
        - Root
```

> **Note**
>
> The Root OU feature is supported only in the V2 version of the manifest file (2021-03-15). If you add **Root** as an OU under `organizational_units`, do not add any other OUs.

Nested OU

CfCT supports listing one or more nested OUs under the `organizational_units` keyword in manifest V2 version (2021-03-15).

A complete path (excluding Root) for the nested OU is required, using a colon as the separator between OUs. For deployment method `scp` or `rcp`, AWS Control Tower deploys the SCPs or RCPs to the last OU in the nested OU path. For deployment method `stack_set`, AWS Control Tower deploys the stack sets to all the accounts under the last OU in the nested OU path.

For example, consider the path OUName1:OUName2:OUName3. The last OU in the path is OUName3. CfCT deploys the SCPs or RCPs to OUName3 and stack sets to all of the accounts directly under OUName3, only.

```
---
region: your-home-region
version: 2021-03-15

resources:

  …truncated…

    deployment_targets:
      organizational_units:
        - OuName1:OUName2:OUName3
```

> **Note**
>
> The nested OU feature is supported only in the V2 version of the manifest file (2021-03-15).

Build your own customizations

To build your own customizations, you can modify the CfCT manifest.yaml file by adding or updating service control policies (SCPs), resource control policies (RCPs), and AWS CloudFormation resources. For resources that must be deployed, you can add or remove accounts and OUs. You can add or modify the templates in the package folders, create your own folders, and reference the templates or folders in the manifest.yaml file.

This section explains the two main parts of building your own customizations:

- how to set up your own configuration package for service control policies
- how to set up your own configuration package for AWS CloudFormation stack sets

Set up a configuration package for SCPs or RCPs

This section explains how to create a configuration package for service control policies (SCPs) or resource control policies (RCPs). The two main parts of this process are (1) prepare the CfCT manifest file, and (2) prepare your folder structure.

Step 1: Edit the manifest.yaml file

Use the sample `manifest.yaml` file as your starting point. Enter all necessary configurations. Add the `resource_file` and `deployment_targets` details.

The following snippet shows the default manifest file.

```
---
region: us-east-1
version: 2021-03-15

resources: []
```

The value for `region` is added automatically during deployment. It must match the Region where you deployed CfCT. This Region must be the same as the AWS Control Tower region.

To add a custom SCP or RCP in the `example-configuration` folder in the zip package stored in the Amazon S3 bucket, open the `example-manifest.yaml` file and begin editing.

```
---
region: your-home-region
version: 2021-03-15

resources:
  - name: test-preventive-controls
    description: To prevent from deleting or disabling resources in member accounts
    resource_file: policies/preventive-controls.json
    deploy_method: scp | rcp
    #Apply to the following OU(s)
    deployment_targets:
      organizational_units: #array of strings
        - OUName1
        - OUName2

…truncated…
```

The following snippet shows an example of a customized manifest file. You can add more than one policy in a single change.

```
---
region: us-east-1
version: 2021-03-15

resources:
  - name: block-s3-public-access
    description: To S3 buckets to have public access
    resource_file: policies/block-s3-public.json
    deploy_method: scp | rcp
    #Apply to the following OU(s)
    deployment_targets:
      organizational_units: #array of strings
        - OUName1
        - OUName2
```

Step 2: Create a folder structure

You can skip this step if you are using an Amazon S3 URL for the resource file and using **parameters** with key/value pairs.

You must include an SCP policy or RCP policy in JSON format to support the manifest, because the manifest file references the JSON file. Ensure that the file paths match the path information provided in the manifest file.

- A *policy* JSON file contains the SCPs or RCPs to be deployed to OUs.

The following snippet shows the folder structure for the sample manifest file.

```
- manifest.yaml
- policies/
   - block-s3-public.json
```

The following snippet is an example of a `block-s3-public.json` policy file.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid":"GuardPutAccountPublicAccessBlock",
          "Effect":"Deny",
          "Action":"s3:PutAccountPublicAccessBlock",
          "Resource":"arn:aws:s3:::*"
       }
    ]
 }
```

Set up a configuration package for AWS CloudFormation StackSets

This section explains how to set up a configuration package for AWS CloudFormation StackSets. The two main parts of this process are: (1) prepare the manifest file, and (2) update the folder structure.

Step 1: Edit the existing manifest file

Add the new AWS CloudFormation StackSets information to the manifest file that you previously edited.

Just for review, the following snippet contains the same customized manifest file that was shown previously to set up a configuration package for SCPs or RCPs. Now you can edit this file further, to include the details about your resources.

```
---
region: us-east-1
version: 2021-03-15

resources:

  - name: block-s3-public-access
    description: To S3 buckets to have public access
    resource_file: policies/block-s3-public.json
    deploy_method: scp | rcp
    #Apply to the following OU(s)
    deployment_targets:
    organizational_units: #array of strings
    - OUName1
    - OUName2
```

The following snippet shows an edited sample manifest file that contains the `resources` details. The order of `resources` determines the execution order for creating `resources` dependencies. You can edit the following example manifest file according to your business requirements.

```
---
region: your-home-region
version: 2021-03-15

…truncated…

resources:
  - name: stackset-1
    resource_file: templates/create-ssm-parameter-keys-1.template
    parameters:
      - parameter_key: parameter-1
        parameter_value: value-1
    deploy_method: stack_set
    deployment_targets:
      accounts: # array of strings, [0-9]{12}
        - account number or account name
        - 123456789123
      organizational_units: #array of strings, ou ids, ou-xxxx
        - OuName1
        - OUName2
    export_outputs:
      - name: /org/member/test-ssm/app-id
        value: $[output_ApplicationId]
    regions:
      - region-name

  - name: stackset-2
    resource_file: s3://bucket-name/key-name
    parameters:
      - parameter_key: parameter-1
        parameter_value: value-1
    deploy_method: stack_set
    deployment_targets:
      accounts: # array of strings, [0-9]{12}
        - account number or account name
        - 123456789123
      organizational_units: #array of strings
        - OuName1
        - OUName2
```

```
regions:
  - region-name
```

The following example shows that you can add more than one AWS CloudFormation resource in the manifest file.

```
---
region: us-east-1
version: 2021-03-15

resources:
  - name: block-s3-public-access
    description: To S3 buckets to have public access
    resource_file: policies/block-s3-public.json
    deploy_method: scp | rcp
    #Apply to the following OU(s)
    deployment_targets:
      organizational_units: #array of strings
        - Custom
        - Sandbox

  - name: transit-network
    resource_file: templates/transit-gateway.template
    parameter_file: parameters/transit-gateway.json
    deploy_method: stack_set
    deployment_targets:
      accounts: # array of strings, [0-9]{12}
        - Prod
        - 123456789123 #Network
      organizational_units: #array of strings
        - Custom
    export_outputs:
      - name: /org/network/transit-gateway-id
        value: $[output_TransitGatewayID]
    regions:
      - us-east-1
```

Step 2: Update the folder structure

When you update the folder structure, you can include all supporting AWS CloudFormation template files and SCP or RCP policy files that are in the manifest file. Verify that the file paths match what is provided in the manifest file.

- A *template* file contains the AWS resources to be deployed in OUs and accounts.
- A *policy* file contains the input parameters used in the template file.

The following example shows the folder structure for the sample manifest file created in Step 1.

```
- manifest.yaml
- policies/
   - block-s3-public.json
- templates/
   - transit-gateway.template
```

The ‘alfred’ helper and the AWS CloudFormation parameter files

CfCT provides you with a mechanism known as the *alfred* helper to get the value for an SSM Parameter Store key that's defined in the AWS CloudFormation template. Using the *alfred* helper, you can use values that are stored in the SSM Parameter Store and without updating the AWS CloudFormation template. For more information, see What is an AWS CloudFormation template? in the *AWS CloudFormation User Guide*.

> **Important**
>
> The *alfred* helper has two limitations. Parameters are available only in the home region of the AWS Control Tower management account. As a best practice, consider working with values that don't change from stack instance to stack instance. When the 'alfred' helper retreives parameters, it chooses a random stack instance from the stack set that exports the variable.

Example

Suppose that you have two AWS CloudFormation stack sets. *Stack set 1* has one stack instance and deploys to one account in one Region. It creates an Amazon VPC and subnets in an availability zone, and the `VPC ID` and `subnet ID` must be passed into *stack set 2* as parameter values. Before the `VPC ID` and `subnet ID` can be passed to *stack set 2*, the `VPC ID` and `subnet ID` must be stored in *stack set 1* using `AWS:::SSM::Parameter`. For more information, see `AWS:::SSM::Parameter` in the *AWS CloudFormation User Guide*.

AWS CloudFormation stack set 1:

In the following snippet, the *alfred* helper can gets value for the `VPC ID` and `subnet ID` from the parameter store and pass them as input to the StackSet state machine.

```
VpcIdParameter:
    Type: AWS::SSM::Parameter
    Properties:
      Name: '/stack_1/vpc/id'
      Description: Contains the VPC id
      Type: String
      Value: !Ref MyVpc

SubnetIdParameter:
    Type: AWS::SSM::Parameter
    Properties:
      Name: '/stack_1/subnet/id'
      Description: Contains the subnet id
      Type: String
      Value: !Ref MySubnet
```

AWS CloudFormation stack set 2:

The snippet shows the parameters that are specified in the AWS CloudFormation stack 2 `manifest.yaml` file.

```
parameters:
      - parameter_key: VpcId
        parameter_value: $[alfred_ssm_/stack_1/vpc/id]
      - parameter_key: SubnetId
        parameter_value: $[alfred_ssm_/stack_1/subnet/id]
```

AWS CloudFormation stack set 2.1:

The snippet shows that you can list `alfred_ssm` properties to support parameters of type *CommaDelimitedList*. For more information, see [Parameters](#) in the *AWS CloudFormation User Guide*.

```
parameters:
      - parameter_key: VpcId # Type: String
        parameter_value: $[alfred_ssm_/stack_1/vpc/id']
      - parameter_key: SubnetId # Type: String
        parameter_value: $[ alfred_ssm_/stack_1/subnet/id']
      - parameter_key: AvailablityZones # Type: CommaDelimitedList
```

```
        parameter_value:
  - "$[alfred_ssm_/availability_zone_1]"
 - "$[alfred_ssm_/availability_zone_2]"
```

JSON schema for the customization package

The JSON schema for the customization package for CfCT is located in the source code repository on GitHub. You can use the schema with many of your favorite development tools, and you may find it helpful for reducing errors when you build your own CfCT `manifest.yaml` file.

Version upgrades for the CfCT manifest

For information about the latest version of *Customizations for AWS Control Tower* (CfCT), see the `CHANGELOG.md` file in the GitHub repository.

Warning

Version 2.2.0 of *Customizations for AWS Control Tower* (CfCT) introduced a CfCT manifest schema (version *2021-03-15*) to align with related AWS service APIs. The manifest schema allows a single manifest.yaml file to manage supported resources (AWS CloudFormation templates, SCPs, and RCPs) through decoupled DevOps workflows.
We strongly recommend that you update the CfCT manifest schema from version *2020-01-01* to version *2021-03-15* or later.
CfCT continues to support version *2021-03-15* and *2020-01-01* of the `manifest.yaml` file. No changes to your existing configuration are required. However, version *2020-01-01* is at **End of Support**. We no longer provide updates or add enhancements to version *2020-01-01*. The Root OU and nested OU features aren't supported in version *2020-01-01*.

Deprecated properties in CfCT manifest version *2021-03-15:*

```
organization_policies
policy_file
apply_to_accounts_in_ou

cloudformation_resources
template_file
```

```
deploy_to_account
deploy_to_ou
ssm_parameters
```

Mandatory CfCT upgrade steps

When you upgrade to the CfCT manifest schema version *2021-03-15* version, here are the changes you must make to update your files. The next sections outline mandatory and recommended changes for the transition.

Organizations polices

1. Move the SCPs or RCPs under **organization_policies** under new property **resources**.
2. Change the **policy_file** property to new property **resource_file.**
3. Change the **apply_to_accounts_in_ou** to new property **deployment_targets**. The OU list should be defined under sub-property **organizational_units**. The **accounts** sub-property is not supported for organizations policies.
4. Add a new property **deploy_method** with the value **scp** or **rcp**.

AWS CloudFormation resources

1. Move the CloudFormation resources under **cloudformation_resources** under new property **resources**.
2. Change the **template_file** property to new property **resource_file**.
3. Change the **deploy_to_ou** to new property **deployment_targets**. The OU list should be defined under sub-property **organizational_units**.
4. Change the **deploy_to_accounts** to new property **deployment_targets**. The account list should be defined under sub-property **accounts**.
5. Change the **ssm_parameters** property to new property **export_outputs**.

Highly recommended CfCT upgrade steps

AWS CloudFormation parameters

1. Change the **parameter_file** property to new property **parameters**.
2. Remove the file path in the value of the **parameter_file** property.

3. Copy the parameter key and parameter value from the existing parameter JSON file into the new format for the **parameters** property. This would help you manage them in the manifest file.

> **Note**
>
> The **parameter_file** property is supported in CfCT manifest version *2021-03-15*.

Networking in AWS Control Tower

AWS Control Tower provides basic support for networking through VPCs.

If the default configuration or capabilities of the AWS Control Tower VPC do not meet your needs, you can use other AWS services to configure your VPC. For more information about how to work with VPCs and AWS Control Tower, see Building a Scalable and Secure Multi-VPC AWS Network Infrastructure.

AWS Control Tower supports IPv4 and IPv6 protocols, through dual-stack IP addresses. For more information, see AWS Control Catalog endpoints and quotas and AWS Control Tower endpoints and quotas.

Related topics

- For information about how AWS Control Tower works when you enroll accounts that have existing VPCs, see Enrolling existing accounts with VPCs.
- With Account Factory, you can provision accounts that include an AWS Control Tower VPC, or you can provision accounts without a VPC. For information about how to delete the AWS Control Tower VPC or configure AWS Control Tower accounts without a VPC, see Walkthrough: Configure AWS Control Tower Without a VPC.
- For information about how to change account settings for VPCs, see the Account Factory documentation on updating an account.
- For more information about working with networking and VPCs in AWS Control Tower, see the section about Networking on the *Related information* page of this *User Guide*.

VPCs and AWS Regions in AWS Control Tower

As a standard part of account creation, AWS creates an AWS-default VPC in every Region, even the Regions you are not governing with AWS Control Tower. This default VPC is not the same as a VPC that AWS Control Tower creates for a provisioned account, but the AWS default VPC in a non-governed Region may be accessible to IAM users.

Adminstrators can enable the Region deny control, so that your end-users do not have permission to connect to a VPC in *a Region that's supported by AWS Control Tower* but outside your governed Regions. To configure the Region deny control, go to the **Landing zone settings** page and select **Modify settings**.

The Region deny control blocks API calls to most services in non-governed AWS Regions. For more information, see Deny access to AWS based on the requested AWS Region..

> **Note**
>
> The Region deny control may not prevent IAM users from connecting to an AWS default VPC in a Region where AWS Control Tower is not supported.

Optionally, you can remove the AWS default VPCs in non-governed Regions. To list the default VPC in a Region you can use a CLI command similar to this example:

```
aws ec2 --region us-west-1 describe-vpcs --filter Name=isDefault,Values=true
```

Overview of AWS Control Tower and VPCs

Here are some essential facts about AWS Control Tower VPCs:

- The VPC created by AWS Control Tower when you provision an account in Account Factory is not the same as the AWS default VPC.
- When AWS Control Tower sets up a new account in a supported AWS Region, AWS Control Tower automatically deletes the default AWS VPC, and it sets up a new VPC configured by AWS Control Tower.
- Each AWS Control Tower account is allowed one VPC that's created by AWS Control Tower. An account can have additional AWS VPCs within the account limit.
- Every AWS Control Tower VPC has three Availability Zones in all Regions except for the US West (N. California) Region,`us-west-1`, and two Availability Zones in `us-west-1`. By default, each Availability Zone is assigned one public subnet and two private subnets. Therefore, in Regions except US West (N. California) each AWS Control Tower VPC contains nine subnets by default, divided across three Availability Zones. In US West (N. California), six subnets are divided across two Availability Zones.
- Each of the subnets in your AWS Control Tower VPC is assigned a unique range, of equal size.
- The number of subnets in a VPC is configurable. For more information about how to change your VPC subnet configuration, see the Account Factory topic.
- Because the IP addresses do not overlap, the six or nine subnets within your AWS Control Tower VPC can communicate with each other in an unrestricted manner.

When working with VPCs, AWS Control Tower makes no distinction at the Region level. Every subnet is allocated from the exact CIDR range that you specify. The VPC subnets can exist in any Region.

Notes

> **Manage VPC costs**
>
> If you set the Account Factory VPC configuration so that public subnets are enabled when provisioning a new account, Account Factory configures VPC to create a NAT Gateway. You will be billed for your usage by Amazon VPC.

> **VPC and control settings**
>
> If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides the control Disallow internet access for an Amazon VPC instance managed by a customer. To avoid enabling internet access for newly provisioned accounts, you must change the setting in Account Factory. For more information, see Walkthrough: Configure AWS Control Tower Without a VPC.

CIDR and Peering for VPC and AWS Control Tower

This section is intended primarily for network administrators. Your company's network administrator usually is the person who selects the overall CIDR range for your AWS Control Tower organization. The network administrator then allocates subnets from within that range for specific purposes.

When you choose a CIDR range for your VPC, AWS Control Tower validates the IP address ranges according to the RFC 1918 specification. Account Factory allows a CIDR block of up to /16 in the ranges of:

- `10.0.0.0/8`
- `172.16.0.0/12`
- `192.168.0.0/16`
- `100.64.0.0/10` (only if your internet provider allows usage of this range)

The /16 delimiter allows up to 65,536 distinct IP addresses.

You can assign any valid IP addresses from the following ranges:

- `10.0.x.x to 10.255.x.x`
- `172.16.x.x – 172.31.x.x`
- `192.168.0.0 – 192.168.255.255` (no IPs outside of `192.168` range)

If the range you specify is outside of these, AWS Control Tower provides an error message.

The default CIDR range is `172.31.0.0/16`.

When AWS Control Tower creates a VPC using the CIDR range you select, it assigns the identical CIDR range to *every VPC* for every account you create within the organizational unit (OU). Due to the default overlap of IP addresses, this implementation does not initially permit peering among any of your AWS Control Tower VPCs in the OU.

Subnets

Within each VPC, AWS Control Tower divides your specified CIDR range evenly into nine subnets (except in US West (N. California), where it is six subnets). None of the subnets within a VPC overlap. Therefore, they all can communicate with each other, within the VPC.

In summary, by default, subnet communication within the VPC is unrestricted. The best practice for controlling communication among your VPC subnets, if needed, is to set up access control lists with rules that define the permitted traffic flow. Use security groups for control of traffic among specific instances. For more information about setting up security groups and firewalls in AWS Control Tower, see Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager.

Peering

AWS Control Tower does not restrict VPC-to-VPC peering for communication across multiple VPCs. However, by default, all AWS Control Tower VPCs have the same default CIDR range. To support peering, you can modify the CIDR range in the settings of Account Factory so that the IP addresses do not overlap.

If you change the CIDR range in the settings of Account Factory, all new accounts that are subsequently created by AWS Control Tower (using Account Factory) are assigned the new CIDR range. The old accounts are not updated. For example, you can create an account, then change the CIDR range and create a new account, and the VPCs allocated to those two accounts can be peered. Peering is possible because their IP address ranges are not identical.

Access AWS Control Tower using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and AWS Control Tower. You can access AWS Control Tower as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to access AWS Control Tower.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for AWS Control Tower.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Considerations for AWS Control Tower

Before you set up an interface endpoint for AWS Control Tower, review Considerations in the *AWS PrivateLink Guide*.

AWS Control Tower supports making calls to all of its API actions through the interface endpoint.

Create an interface endpoint for AWS Control Tower

You can create an interface endpoint for AWS Control Tower using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create an interface endpoint for AWS Control Tower using the following service names:

```
com.amazonaws.region.controltower
com.amazonaws.region.controltower-fips
```

If you enable private DNS for the interface endpoint, you can make API requests to AWS Control Tower using its default Regional DNS name. For example, `controltower.us-east-1.amazonaws.com`.

Create an endpoint policy for your interface endpoint

An endpoint policy is an IAM resource that you can attach to an interface endpoint. The default endpoint policy allows full access to AWS Control Tower through the interface endpoint. To control the access allowed to AWS Control Tower from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals that can perform actions (AWS accounts, IAM users, and IAM roles).
- The actions that can be performed.
- The resources on which the actions can be performed.

For more information, see Control access to services using endpoint policies in the *AWS PrivateLink Guide*.

Example: VPC endpoint policy for AWS Control Tower actions

The following is an example of a custom endpoint policy. When you attach this policy to your interface endpoint, it grants access to the listed AWS Control Tower actions for all principals on all resources.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "controltower:ListEnabledControls",
            "controltower:ListLandingZones"
         ],
         "Resource":"*"
      }
   ]
}
```

> **Note**
>
> For a complete list of AWS Control Tower API operations, see the AWS Control Tower API Reference.

Required roles and permissions

AWS Control Tower uses IAM roles to help manage access to resources.

For general information about roles, see User groups, roles, and permission sets.

About permissions

- For information about IAM groups and their permissions in AWS Control Tower, see IAM Identity Center groups for AWS Control Tower.
- For information about permissions required to provision accounts, see Permissions required for accounts.
- For information about console permissions required for AWS Control Tower, see Permissions required to use the AWS Control Tower console.

About roles

- For information about how to create a role, including permissions designed for programmatic access, see Create roles and assign permissions, and Programmatic roles and trust relationships for the AWS Control Tower audit account.
- For information about other roles that AWS Control Tower uses to manage your accounts, see Using identity-based policies (IAM policies) for AWS Control Tower, and the Managed policies for AWS Control Tower.
- For information about AWS Control Tower and AWS Config roles, see AWS Control Tower ConfigRecorderRole.
- For information about roles that AWS Control Tower uses to aggregate AWS Config information for your accounts, see How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts.
- For information about how to protect your resources as you are assigning roles and permissions, see Optional conditions for your role trust relationships, Optionally configure AWS KMS keys, and Prevent cross-service impersonation.
- For specific information about automated account provisioning in AWS Control Tower with IAM roles, see Automated Account Provisioning with IAM Roles.
- To view the policy that protects the AWS Config SNS topic, see The AWS Config SNS topic policy.

How AWS Control Tower works with roles to create and manage accounts

In general, roles are a part of identity and access management (IAM) in AWS. For general information about IAM and roles in AWS, see the IAM roles topic in the *AWS IAM User Guide*.

Roles and account creation

AWS Control Tower creates a customer's account by calling the `CreateAccount` API of AWS Organizations. When AWS Organizations creates this account, it creates a role within that account, which AWS Control Tower names by passing in a parameter to the API. The name of the role is `AWSControlTowerExecution`.

AWS Control Tower takes over the `AWSControlTowerExecution` role for all accounts created by Account Factory. Using this role, AWS Control Tower *baselines* the account and applies mandatory (and any other enabled) controls, which results in creation of other roles. These roles in turn are used by other services, such as AWS Config.

> **Note**
>
> To *baseline* an account is to set up its resources, which include Account Factory templates, sometimes referred to as *blueprints*, and controls. The baselining process also sets up the centralized logging and security audit roles on the account, as part of deploying the templates. AWS Control Tower baselines are contained in the roles that you apply to every enrolled account.

For more information about accounts and resources, see About AWS accounts in AWS Control Tower.

The AWSControlTowerExecution role, explained

The `AWSControlTowerExecution` role must be present in all enrolled accounts. It allows AWS Control Tower to manage your individual accounts and report information about them to your Audit and Log Archive accounts.

The `AWSControlTowerExecution` role can be added into an account in several ways, as follows:

- For accounts in the Security OU (sometimes called *core accounts*), AWS Control Tower creates the role at the time of initial AWS Control Tower setup.
- For an Account Factory account created through the AWS Control Tower console, AWS Control Tower creates this role at the time of account creation.
- For a single account enrollment, we ask customers to manually create the role and then enroll the account in AWS Control Tower.
- When extending governance to an OU, AWS Control Tower uses the **StackSet-AWSControlTowerExecutionRole** to create the role in all accounts in that OU.

Purpose of the `AWSControlTowerExecution` role:

- `AWSControlTowerExecution` allows you to create and enroll accounts, automatically, with scripts and Lambda functions.
- `AWSControlTowerExecution` helps you configure your organizations's logging, so that all the logs for every account are sent to the logging account.
- `AWSControlTowerExecution` allows you to enroll an individual account in AWS Control Tower. First, you must add the `AWSControlTowerExecution` role to that account. For steps on how to add the role, see Manually add the required IAM role to an existing AWS account and enroll it.

How the `AWSControlTowerExecution` role works with OUs:

The `AWSControlTowerExecution` role ensures that your selected AWS Control Tower controls apply automatically to every individual account, in each OU, in your organization, as well as to every new account you create in AWS Control Tower. As a result:

- You can provide compliance and security reports more easily, based on the auditing and logging features embodied by AWS Control Tower controls.
- Your security and compliance teams can verify that all requirements are met, and that no organizational drift has occurred.

For more information about drift, see Detect and resolve drift in AWS Control Tower.

To summarize, the `AWSControlTowerExecution` role and its associated policy gives you flexible control of security and compliance across your entire organization. Therefore, breaches of security or protocol are less likely to occur.

Optional conditions for your role trust relationships

You can impose conditions in your role trust policies, to restrict the accounts and resources that interact with certain roles in AWS Control Tower. We strongly recommend that you restrict access to the `AWSControlTowerAdmin` role, because it allows wide access permissions.

To help prevent an attacker from gaining access to your resources, manually edit your AWS Control Tower trust policy to add at least one `aws:SourceArn` or `aws:SourceAccount` conditional to the policy statement. As a security best practice, we strongly recommend adding the `aws:SourceArn` condition, because it is more specific than `aws:SourceAccount`, limiting access to a specific account and a specific resource.

If you don't know the full ARN of the resource, or if you are specifying multiple resources, you can use the `aws:SourceArn` condition with wildcards (*) for the unknown portions of the ARN. For example, `arn:aws:controltower:*:123456789012:*` works if you don't wish to specify a Region.

The following example demonstrates the use of the `aws:SourceArn` IAM condition with your IAM role trust polices. Add the condition in your trust relationship for the **AWSControlTowerAdmin** role, because the AWS Control Tower service principal interacts with it.

As shown in the example, the source ARN is of the format:
`arn:aws:controltower:${HOME_REGION}:${CUSTOMER_AWSACCOUNT_id}:*`

Replace the strings `${HOME_REGION}` and `${CUSTOMER_AWSACCOUNT_id}` with your own home Region and account ID of the calling account.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "controltower.amazonaws.com"
        ]
      },
```

```
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:controltower:us-west-2:012345678901:*"
        }
      }
    }
  ]
}
```

In the example, the Source ARN designated as `arn:aws:controltower:us-west-2:012345678901:*` is the only ARN allowed to perform the `sts:AssumeRole` action. In other words, only users who can sign in to the account ID `012345678901`, in the `us-west-2` Region, are allowed to perform actions that require this specific role and trust relationship for the AWS Control Tower service, designated as `controltower.amazonaws.com`.

The next example shows the `aws:SourceAccount` and `aws:SourceArn` conditions applied to the role trust policy.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "controltower.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "012345678901"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:controltower:us-west-2:012345678901:*"
        }
      }
    }
```

```
    ]
}
```

The example illustrates the `aws:SourceArn` condition statement, with an added `aws:SourceAccount` condition statement. For more information, see Prevent cross-service impersonation.

For general information about permission policies in AWS Control Tower see Manage access to resources.

Recommendations:

We recommend that you add conditions to the roles that AWS Control Tower creates, because those roles are directly assumed by other AWS services. For more information, see the example for **AWSControlTowerAdmin**, shown previously in this section. For the AWS Config recorder role, we recommend adding the `aws:SourceArn` condition, specifying the Config recorder ARN as the permitted source ARN.

For roles such as **AWSControlTowerExecution** or the other programmatic roles that can be assumed by the AWS Control Tower Audit account in all managed accounts, we recommend that you add the `aws:PrincipalOrgID` condition to the trust policy for these roles, which validates that the principal accessing the resource belongs to an account in the correct AWS organization. Do not add the `aws:SourceArn` condition statement, because it will not work as expected.

> **Note**
>
> In case of drift, it is possible that an AWS Control Tower role may be reset under certain circumstances. It is recommended that you re-check the roles periodically, if you have customized them.

How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts

The AWS Control Tower management account creates an organization-level aggregator, which assists in detecting external AWS Config rules, so that AWS Control Tower does not need to gain access to unmanaged accounts. The AWS Control Tower console shows you how many externally created AWS Config rules you have for a given account. You can view details about those external rules in the **External Config Rule Compliance** tab of the **Account details** page.

To create the aggregator, AWS Control Tower adds a role with the permissions required to describe an organization and list the accounts under it. The `AWSControlTowerConfigAggregatorRoleForOrganizations` role requires the `AWSConfigRoleForOrganizations` managed policy and a trust relationship with `config.amazonaws.com`.

Here is the IAM policy (JSON artifact) attached to the role:

JSON

```
{
    "Version": "2012-10-17",
      "Statement": [
       {
        "Effect": "Allow",
        "Action": [
          "organizations:ListAccounts",
          "organizations:DescribeOrganization",
          "organizations:ListAWSServiceAccessForOrganization"
         ],
       "Resource": "*"
      }
    ]
  }
```

Here is the `AWSControlTowerConfigAggregatorRoleForOrganizations` trust relationship:

```
{
    "Version": "2012-10-17",
      "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
        "Service": "config.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
```

```
    ]
  }
}
```

To deploy this functionality in the management account, the following permissions are added in the managed policy `AWSControlTowerServiceRolePolicy`, which is used by the `AWSControlTowerAdmin` role when it creates the AWS Config aggregator:

JSON

```
{
  "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "config:PutConfigurationAggregator",
          "config:DeleteConfigurationAggregator",
          "iam:PassRole"
          ],
        "Resource": [
          "arn:aws:iam::123456789012:role/service-role/
AWSControlTowerConfigAggregatorRoleForOrganizations",
          "arn:aws:config:us-east-1:123456789012:config-aggregator/"
          ]
        },
      {
        "Effect": "Allow",
        "Action": "organizations:EnableAWSServiceAccess",
        "Resource": "*"
      }
    ]
}
```

New resources created: `AWSControlTowerConfigAggregatorRoleForOrganizations` and `aws-controltower-ConfigAggregatorForOrganizations`

When you are ready, you can enroll accounts individually, or enroll them as a group by registering an OU. When you've enrolled an account, if you create a rule in AWS Config, AWS Control Tower detects the new rule. The aggregator shows the number of external rules and provides a link to the

AWS Config console where you can view the details of each external rule for your account. Use the information in the AWS Config console and the AWS Control Tower console to determine whether you have the appropriate controls enabled for the account.

Programmatic roles and trust relationships for the AWS Control Tower audit account

You can sign into the audit account and assume a role to review other accounts programmatically. The audit account does not allow you to log in to other accounts manually.

The audit account gives you programmatic access to other accounts, by means of some roles that are granted to AWS Lambda functions only. For security purposes, these roles have *trust relationships* with other roles, which means that the conditions under which the roles can be utilized are strictly defined.

The AWS Control Tower stack `StackSet-AWSControlTowerBP-BASELINE-ROLES` creates these programmatic-only, cross-account IAM roles in the audit account:

- **aws-controltower-AdministratorExecutionRole**
- **aws-controltower-ReadOnlyExecutionRole**

The AWS Control Tower stack `StackSet-AWSControlTowerSecurityResources` creates these programmatic-only, cross-account IAM roles in the audit account:

- **aws-controltower-AuditAdministratorRole**
- **aws-controltower-AuditReadOnlyRole**

`ReadOnlyExecutionRole:` Note that this role allows the audit account to read objects in Amazon S3 buckets across the entire organization (in contrast to the `SecurityAudit` policy, which allows for metadata access only).

aws-controltower-AdministratorExecutionRole:

- Has administrator permissions
- Cannot be assumed from the console
- Can be assumed only by a role in the audit account – the `aws-controltower-AuditAdministratorRole`

The following artifact shows the trust relationship for `aws-controltower-AdministratorExecutionRole`. The placeholder number `012345678901` will be replaced by the `Audit_acct_ID` number for your audit account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/aws-controltower-AuditAdministratorRole"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

aws-controltower-AuditAdministratorRole:

- Can be assumed by the AWS Lambda service only
- Has permission to perform read (Get) and write (Put) operations on Amazon S3 objects with names that start with the string **log**

Attached policies:

1. **AWSLambdaExecute** – AWS managed policy

2. **AssumeRole-aws-controltower-AuditAdministratorRole** – inline policy – Created by AWS Control Tower, artifact follows.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
 {
 "Action": [
   "sts:AssumeRole"
   ],
 "Resource": [
```

```
            "arn:aws:iam::*:role/aws-controltower-AdministratorExecutionRole"
          ],
        "Effect": "Allow"
        }
          ]
      }
```

The following artifact shows the trust relationship for `aws-controltower-AuditAdministratorRole`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

aws-controltower-ReadOnlyExecutionRole:

- Cannot be assumed from the console
- Can be assumed only by another role in the audit account – the `AuditReadOnlyRole`

The following artifact shows the trust relationship for `aws-controltower-ReadOnlyExecutionRole`. The placeholder number `012345678901` will be replaced by the `Audit_acct_ID` number for your audit account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/aws-controltower-AuditReadOnlyRole "
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

aws-controltower-AuditReadOnlyRole:

- Can be assumed by the AWS Lambda service only
- Has permission to perform read (Get) and write (Put) operations on Amazon S3 objects with names that start with the string **log**

Attached policies:

1. **AWSLambdaExecute** – AWS managed policy

2. **AssumeRole-aws-controltower-AuditReadOnlyRole** – inline policy – Created by AWS Control Tower, artifact follows.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
 {
 "Action": [
  "sts:AssumeRole"
 ],
 "Resource": [
  "arn:aws:iam::*:role/aws-controltower-ReadOnlyExecutionRole"
 ],
 "Effect": "Allow"
   }
  ]
}
```

The following artifact shows the trust relationship for `aws-controltower-AuditAdministratorRole`:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Automated Account Provisioning With IAM Roles

To configure Account Factory accounts in a more automated way, you can create Lambda functions in the AWS Control Tower management account, which [assumes the **AWSControlTowerExecution** role](#) in the member account. Then, using the role, the management account performs the desired configuration steps in each member account.

If you're provisioning accounts using Lambda functions, the identity that will perform this work must have the following IAM permissions policy, in addition to `AWSServiceCatalogEndUserFullAccess`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSControlTowerAccountFactoryAccess",
            "Effect": "Allow",
            "Action": [
                "sso:GetProfile",
                "sso:CreateProfile",
                "sso:UpdateProfile",
                "sso:AssociateProfile",
                "sso:CreateApplicationInstance",
                "sso:GetSSOStatus",
                "sso:GetTrust",
                "sso:CreateTrust",
                "sso:UpdateTrust",
```

```
                "sso:GetPeregrineStatus",
                "sso:GetApplicationInstance",
                "sso:ListDirectoryAssociations",
                "sso:ListPermissionSets",
                "sso:GetPermissionSet",
                "sso:ProvisionApplicationInstanceForAWSAccount",
                "sso:ProvisionApplicationProfileForAWSAccountInstance",
                "sso:ProvisionSAMLProvider",
                "sso:ListProfileAssociations",
                "sso-directory:ListMembersInGroup",
                "sso-directory:AddMemberToGroup",
                "sso-directory:SearchGroups",
                "sso-directory:SearchUsers",
                "sso-directory:CreateUser",
                "sso-directory:DescribeGroups",
                "sso-directory:DescribeDirectory",
                "sso-directory:GetUserPoolInfo",
                "controltower:CreateManagedAccount",
                "controltower:DescribeManagedAccount",
                "controltower:DeregisterManagedAccount",
                "s3:GetObject",
                "organizations:describeOrganization",
                "sso:DescribeRegisteredRegions"
            ],
            "Resource": "*"
        }
    ]
}
```

The permissions `sso:GetPeregrineStatus`, `sso:ProvisionApplicationInstanceForAWSAccount`, `sso:ProvisionApplicationProfileForAWSAccountInstance`, and `sso:ProvisionSAMLProvide` are required by AWS Control Tower Account Factory to interact with AWS IAM Identity Center.

Resources in AWS Control Tower

- For general information about resource ownership in AWS Control Tower, see Overview of managing access permissions to your AWS Control Tower resources.
- For information about resources that AWS Control Tower creates in the shared accounts, see About the shared accounts.
- For information about resources that AWS Control Tower creates when it provisions an account through Account Factory, see Resource Considerations for Account Factory.
- To view details about the AWS resource types that are defined by AWS Control Tower, for use with the AWS Control Tower APIs, see the AWS Control Tower resource type reference in the *AWS CloudFormation User Guide*.

How AWS Regions Work With AWS Control Tower

AWS Control Tower is supported in the following AWS Regions:

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Canada (Central)
- Asia Pacific (Sydney)
- Asia Pacific (Singapore)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Stockholm)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Tokyo)
- Europe (Paris)
- South America (São Paulo)
- US West (N. California)
- Asia Pacific (Hong Kong)
- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- Europe (Milan)
- Africa (Cape Town)
- Middle East (Bahrain)
- Israel (Tel Aviv)
- Middle East (UAE)
- Europe (Spain)
- Asia Pacific (Hyderabad)

- Europe (Zurich)
- Asia Pacific (Melbourne)
- Canada West (Calgary)
- Malaysia (Kuala Lumpur)
- Asia Pacific (Thailand)
- Mexico (Central)
- Asia Pacific (Taipei)

About your home Region

When you create a landing zone, the Region that you're using for access to the AWS Management console becomes your home AWS Region for AWS Control Tower. During the creation process, some resources are provisioned in the home Region. Other resources, such as OUs and AWS accounts, are global.

After you've selected a home Region, you cannot change it.

Controls and Regions

Currently, all preventive controls work globally. Detective and proactive controls, however, only work in Regions where AWS Control Tower is supported. For more information about the behavior of controls when you activate AWS Control Tower in a new Region, see Configure your AWS Control Tower Regions.

Configure your AWS Control Tower Regions

This section describes the behavior you can expect when you extend your AWS Control Tower landing zone into a new AWS Region, or remove a Region from your landing zone configuration. Generally, this action is performed through the **Update** function of the AWS Control Tower console.

> **Note**
>
> We recommend that you avoid expanding your AWS Control Tower landing zone into AWS Regions in which you do not require your workloads to run. Opting out of a Region does not prevent you from deploying resources in that Region, but those resources will remain outside of AWS Control Tower governance.

During configuration of a new Region, AWS Control Tower updates the landing zone, which means that it *baselines* your landing zone —

- to operate actively in all newly-selected Regions, and
- to cease governing resources in deselected Regions.

Individual accounts within your organizational units (OUs) that are managed by AWS Control Tower are not updated as part of this landing zone update process. Therefore, you must update your accounts by re-registering your OUs.

When configuring your AWS Control Tower Regions, be aware of the following recommendations and limitations:

- Select Regions in which you plan to host AWS resources or workloads.
- Opting out of a Region does not prevent you from deploying resources in that Region, but those resources will remain outside of AWS Control Tower governance.

When you configure your landing zone for new Regions, AWS Control Tower detective controls adhere to the following rules:

- *What exists stays the same.* Control behavior, detective as well as preventive, is unchanged for existing accounts, in existing OUs, in existing Regions.
- *You can't apply new detective controls to existing OUs containing accounts that are not updated.* When you've configured your AWS Control Tower landing zone into a new Region (by updating your landing zone), you must update existing accounts in your existing OUs before you can enable new detective controls on those OUs and accounts.
- *Your existing detective controls begin working in the newly configured Regions as soon as you update the accounts.* When you update your AWS Control Tower landing zone to configure new Regions and then update an account, the detective controls that already are enabled on the OU will begin working on that account in the newly configured Regions.

Configure AWS Control Tower Regions

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower
2. In the left-pane navigation menu, choose **Landing zone settings**.

3. On the **Landing zone settings** page,in the **Details** section, choose the **Modify settings** button in the upper right. You are directed to the update landing zone workflow, because governing new Regions, or removing Regions from governance, requires you to update to the latest landing zone version.
4. Under **Additional AWS Regions for governance**, search for the Regions you want to govern (or stop governing). The **State** column indicates which Regions you currently govern, and which ones you don't.
5. Select the checkbox for each additional Region to govern. Deselect the checkbox for each Region from which you are removing governance.

> **Note**
>
> If you opt not to govern a Region, you can still deploy resources in that Region, but those resources will remain outside of AWS Control Tower governance.

6. Complete the rest of the workflow, then choose **Update landing zone**.
7. When the landing zone setup completes, **Re-register** the OUs to update the accounts in your new Regions. For more information, see When to update AWS Control Tower OUs and accounts.

An alternative method of provisioning or updating individual accounts after configuring new Regions is by using the API framework of Service Catalog and the AWS CLI to update the accounts in a batch process. For more information, see Provision and update accounts using automation.

Avoid mixed governance when configuring Regions

It is important to update all accounts in an OU after you extend AWS Control Tower governance to a new AWS Region, and after you remove AWS Control Tower governance from a Region.

Mixed governance is an undesirable situation that can occur if the controls governing an OU are not a complete match to the controls governing each account within an OU. Mixed governance occurs in an OU if accounts are not updated after AWS Control Tower extends governance to a new AWS Region, or removes governance.

In this situation, certain accounts within an OU may have different controls applied in different Regions, when compared to other accounts in the OU, or when compared to the landing zone's overall governance posture.

In an OU with mixed governance, if you provision a new account, that new account receives the same (updated) Region and OU governance posture as the landing zone. However, existing accounts that are not yet updated do not receive the updated Region governance posture.

In general, mixed governance may create contradictory or inaccurate status indicators in the AWS Control Tower console. For example, during mixed governance, opt-in Regions are shown with **Not governed** status, in registered OUs, for accounts that are not yet updated.

> **Note**
>
> AWS Control Tower does not permit controls to be enabled during a state of mixed governance.

Behavior of controls during mixed governance

- During mixed governance, AWS Control Tower cannot consistently deploy controls that are based on AWS Config rules (that is, detective controls) in Regions that the OU already shows as **Governed**, because some accounts in the OU have not been updated. You may receive a `FAILED_TO_ENABLE` error message.
- During mixed governance, if you extend the landing zone's governance to an opt-in Region while any account in the OU has not yet been updated, the `EnableControl` API operation on the OU fails for detective and proactive controls. You will receive a `FAILED_TO_ENABLE` error message, because non-updated member accounts within the OU have not yet been opted into those Regions.
- During mixed governance, controls that are part of the **Security Hub Service-managed Standard: AWS Control Tower** cannot report compliance accurately in Regions where there is a mismatch between the landing zone configuration and the accounts that are not updated.
- Mixed governance does not change the behavior of SCP-based controls (preventive controls), which apply uniformly to every account in an OU, in every governed Region.

> **Note**
>
> Mixed governance is not the same as drift, and it is not reported as drift.

To repair mixed governance

- Choose **Update account** for each account in the OU that shows **Update available** status on the **Organizations** page in the console.
- Choose **Re-Register OU** on the **Organizations** page, which automatically updates all accounts in the OU, for OUs with fewer than 1000 accounts.

Considerations for activating AWS opt-in Regions

Although most AWS Regions are active by default for your AWS account, certain Regions are activated only when you manually select them. This document refers to those Regions as *opt-in Regions*. In contrast, Regions that are active by default, as soon as your AWS account is created, are referred to as *commercial Regions*, *default Regions*, or simply, *Regions*.

The term *opt-in* has a historical basis. Any AWS Regions introduced after March 20, 2019 are deployed as opt-in Regions. In an opt-in Region, your account is not enabled within that Region —and your identity is not replicated to the Region—unless you choose to opt into use of that Region. All of the data managed through the IAM service is considered identity data, including users, groups, roles, policies, identity providers, their associated data (for example, X.509 signing certificates or context-specific credentials), and other account-level settings, such as password policy and the account alias.

You can activate opt-in Regions automatically during landing zone setup, by selecting them. Your landing zone becomes active in all selected Regions.

If you choose to select an opt-in Region as your AWS Control Tower home Region, activate it first by following the steps in [Enabling a Region](), when signed in to the AWS Management Console. To bring your own existing Log Archive and Audit accounts from an opt-in Region, manually activate that Region first.

The AWS opt-in Regions include several Regions in which AWS Control Tower is available:

- Asia Pacific (Hong Kong) Region, ap-east-1
- Asia Pacific (Jakarta) Region, ap-southeast-3
- Europe (Milan) Region, eu-south-1
- Africa (Cape Town) Region, af-south-1
- Middle East (Bahrain) Region, me-south-1

- Israel (Tel Aviv), il-central-1
- Middle East (UAE) Region, me-central-1
- Europe (Spain) Region, eu-south-2
- Asia Pacific (Hyderabad) Region, ap-south-2
- Europe (Zurich) Region, eu-central-2
- Asia Pacific (Melbourne) Region, ap-southeast-4
- Canada West (Calgary) Region, ca-west-1
- Asia Pacific (Thailand), ap-southeast-7
- Mexico (Central), mx-central-1
- Asia Pacific (Taipei), ap-east-2

AWS Control Tower has some controls that work differently in the opt-in Regions than in the default Regions (other commercial Regions). For more information, see Control limitations. Here are some considerations to keep in mind as you deploy workloads into opt-in Regions.

Governing or activating?

Remember that governing a Region is an action that you can select from the AWS Control Tower console, so that controls can be applied in the Region. Activating or deactivating an opt-in Region is a different action that you can choose in the AWS console, which opens the Region to your account, so that you can deploy resources and workloads in the Region.

Behavioral considerations

- If you choose to govern opt-in Regions, we recommend that you do not deactivate (opt-out of) any of your governed opt-in Regions, because it can lead to failure of your workloads. AWS Control Tower does not allow deactivation of a governed Region from within the AWS Control Tower console, but be sure that you do not deactivate governed Regions from a source outside of AWS Control Tower, such as the AWS Billing console or AWS SDK.
- When AWS Control Tower extends governance to an opt-in Region, it activates (opts-in) to the Region in all member accounts. When you remove a Region from governance, AWS Control Tower does not deactivate (opt-out of) the Region in the member accounts.
- During Region deselection, AWS Control Tower skips removing resources from an opt-in Region if that Region was deactivated manually for an account from a source outside AWS Control

Tower, such as the AWS Billing console or AWS SDK. We recommend that you remove resources from the Regions you've deactivated, or you may receive unexpected billing charges for those resources.

- If your landing zone is decommissioned, AWS Control Tower cleans up resources in all the governed Regions, including the opt-in Regions. However, AWS Control Tower does not deactivate the opt-in Regions. You can deactivate the opt-in Regions as an additional step after decommissioning.
- If your home Region is an opt-in Region, and if you intend to enroll existing accounts as your Log Archive and Audit accounts, you must manually activate the opt-in Region before you can select it as the home Region for your landing zone. See Enabling a Region.
- If AWS Control Tower is set up with an opt-in Region as your home Region, and if you visit the AWS Control Tower service from the AWS console in any other Region, the console does not redirect you automatically to the home Region.
- The underlying API has capacity limits, which may increase latency from a few minutes to many hours, depending on the number of Regions, accounts, and service load. As a best practice, opt-in only to those the AWS Regions where you will run workloads, and opt-in one Region at a time.

Important limitations for governance and accounts

- If 16 or more commercial Regions where AWS Control Tower is available are governed, including opt-in Regions, the upper limit on the number of accounts per organizational unit (OU), is reduced, when registering an OU. For more information, see Limitations based on underlying AWS services.

Configure the Region deny control

AWS Control Tower offers two Region deny controls. One control, `GRREGIONDENY`, when activated, applies to the entire landing zone. Another control, `CTMULTISERVICEPV1`, when activated, can apply to specific OUs that you specify. For more information see Deny access to AWS based on the requested AWS Region and Region deny control applied to the OU.

Considerations about the Region deny control for the landing zone

The Region deny control, `GRREGIONDENY` is unique, because it applies to the landing zone as a whole, rather than to any specific OU. To configure the Region deny control, go to the **Landing zone settings** page and select **Modify settings**.

- This setting can be changed at a later time.
- When enabled, this control applies to all registered OUs.
- This control cannot be configured for individual OUs.

Note

Before you enable the Region deny control, be sure that you do not have existing resources in these Regions, because you will not have access to your resources after you apply the control. While the control is enabled, you will not be able to deploy resources in the denied Regions.

When you enable the control, it applies to all registered, top-level OUs in your hierarchy, and it is inherited by OUs lower in the chain. When you remove the control, it is removed on all registered OUs, all non-governed Regions in AWS Control Tower remain in a **Not governed** status, and you can deploy resources in Regions outside of AWS Control Tower availability.

Exceptions

You cannot deny access to your home Region. Certain global AWS services, such as IAM and AWS Organizations, are exempt from the Region deny control. To learn more, see Deny access to AWS based on the requested AWS Region.

- Full control name: **Deny access to AWS based on the requested AWS Region**
- Control description: Disallows access to unlisted operations in global and regional services outside of the specified Regions.
- This is an elective control with preventive guidance.

To view the template for the Region deny control SCP, see Deny access to AWS based on the requested AWS Region in the *AWS Control Tower Control reference*. The AWS Control Tower SCP is similar to the SCP for AWS Organizations, but not identical.

You can determine Regional service endpoints on the Regional services page.

Considerations for the OU-level Region deny control

The primary consideration about the OU-level Region deny control is to determine how it will interact with the landing zone Region deny control, if both are activated. For more information, see Region deny control applied to the OU.

You also may wish to review Configure the Region deny control.

Provision and manage accounts in AWS Control Tower

This chapter includes an overview and procedures for provisioning and managing member accounts in your AWS Control Tower landing zone.

It also includes an overview and procedures for enrolling an existing AWS account into AWS Control Tower.

For more information about accounts in AWS Control Tower, see About AWS accounts in AWS Control Tower. For information about enrolling multiple acounts into AWS Control Tower, see Register an existing organizational unit with AWS Control Tower.

Note

You can perform up to five (5) account-related operations concurrently, including provisioning, updating, and enrolling.

Methods of provisioning

AWS Control Tower provides several methods for creating and updating member accounts. Some methods are primarily console-based, and some methods are primarily automated.

Overview

The standard way to create member accounts is through Account Factory, a console-based product that's part of the Service Catalog. If your landing zone is not in a state of drift, you can use **Create account** as a method to add new accounts from the console, as well as **Enroll account** to enroll existing AWS accounts into AWS Control Tower.

With Account Factory, you can provision basic accounts, by relying on the AWS Control Tower default settings. You also can provision customized accounts that meet requirements for specialized use cases.

Account Factory Customization (AFC) is a way of provisioning customized accounts from the AWS Control Tower console, and it automates the customization and deployment of your accounts. It allows console-based, automated provisioning, after some one-time setup steps, which eliminates the need to write scripts or set up pipelines. For more information, see Customize accounts with Account Factory Customization (AFC).

Console-based methods:

- Through the Account Factory console that is part of AWS Service Catalog, for basic or customized accounts. Review Provision and manage accounts with Account Factory for details and instructions.
- Through the **Enroll account** feature within AWS Control Tower, if your landing zone is not in a state of drift. See Enroll an existing account.
- In the AWS Control Tower console, you can use Account Factory to create, update, or enroll up to five accounts at the same time.

Automated methods:

- **Lambda code:** From your AWS Control Tower landing zone's management account, using Lambda code and appropriate IAM roles. See Automated Account Provisioning with IAM Roles.
- **Terraform:** From the AWS Control Tower Account Factory for Terraform (AFT), which relies on Account Factory and a GitOps model to allow automation of account provisioning and updating. See Provision accounts with AWS Control Tower Account Factory for Terraform (AFT) .
- **Account Factory customization in the AWS Control Tower console:** After the setup steps, future provisioning of customized accounts requires no additional configuration or pipeline maintenance. Accounts are provisioned by means of a AWS Service Catalog product called a *blueprint*. A blueprint can use AWS CloudFormation templates, or Terraform templates.

> **Note**
>
> AWS CloudFormation blueprints can deploy resources to multiple Regions. Terraform blueprints can deploy resources to a single Region only. By default, that is the home Region.

What happens when AWS Control Tower creates an account

New accounts in AWS Control Tower are created and then provisioned by an interaction among AWS Control Tower, AWS Organizations, and AWS Service Catalog. For steps to enroll an existing AWS account using the AWS Control Tower console, see Enroll an existing account.

Behind the scenes of account creation

1. You initiate the request, for example, from the AWS Control Tower Account Factory page, or directly from the AWS Service Catalog console, or by calling the Service Catalog `ProvisionProduct` API.
2. AWS Service Catalog calls AWS Control Tower.
3. AWS Control Tower begins a workflow, which as a first step calls the AWS Organizations `CreateAccount` API.
4. After AWS Organizations creates the account, AWS Control Tower completes the provisioning process by applying blueprints and controls.
5. Service Catalog continues to poll AWS Control Tower to check for completion of the provisioning process.
6. When the workflow in AWS Control Tower is complete, Service Catalog finalizes the account's state and informs you (the requester) of the result.

Permissions required for accounts

The permissions required for each method of provisioning and updating accounts are discussed in each section, respectively. With the appropriate user group permissions, provisioners can specify standardized baselines and network configurations for any accounts in their organization.

> **Note**
>
> When provisioning an account, the account requester always must have the `CreateAccount` and the `DescribeCreateAccountStatus` permissions. This permission set is part of the Admin role, and it is given automatically when a requester assumes the Admin role. If you delegate permission to provision accounts, you may need to add these permissions directly for the account requestors.

When you create accounts from the AWS Control Tower console with Account Factory, you must be signed into an account with an IAM user that has the `AWSServiceCatalogEndUserFullAccess` policy enabled, along with permissions to use the AWS Control Tower console, and you cannot be signed in as the **Root** user.

For general information about permissions required in AWS Control Tower, see Using identity-based policies (IAM policies) for AWS Control Tower. For information about roles and accounts in AWS Control Tower, see Roles and accounts.

Security for your accounts

You can find guidance about best practices to protect the security of your AWS Control Tower management account and member accounts in the AWS Organizations documentation.

- Best practices for the management account
- Best practices for member accounts

About AWS accounts in AWS Control Tower

An AWS account is the container for all your owned resources. These resources include the AWS Identity and Access Management (IAM) identities accepted by the account, which determine who has access to that account. IAM identities can include users, groups, roles, and more. For more information about working with IAM, users, roles, and policies in AWS Control Tower, see Identity and access management in AWS Control Tower.

Resources and account creation time

When AWS Control Tower creates or enrolls an account, it deploys the minimum necessary resource configuration for the account, including resources in the form of Account Factory templates and other resources in your landing zone. These resources may include IAM roles, AWS CloudTrail trails, Service Catalog provisioned products, and IAM Identity Center users. AWS Control Tower also deploys resources, as required by the control configuration, for the organizational unit (OU) in which the new account is destined to become a member account.

AWS Control Tower orchestrates the deployment of these resources on your behalf. It may require several minutes per resource to complete the deployment, so consider the total time before you create or enroll an account. For more information about managing resources in your accounts, see Guidance for creating and modifying AWS Control Tower resources.

Considerations for bringing existing security or logging accounts

Before accepting an AWS account as a security or logging account, AWS Control Tower checks the account for resources that conflict with AWS Control Tower requirements. For example, you may have a logging bucket with the same name that AWS Control Tower requires. Also, AWS Control Tower validates that the account can provision resources; for example, by ensuring that AWS Security Token Service (AWS STS) is enabled, that the account is not suspended, and that AWS Control Tower has permission to provision resources within the account.

AWS Control Tower does not remove any existing resources in the logging and security accounts that you provide. However, if you choose to enable the AWS Region deny capability, the Region deny control prevents access to resources in denied Regions.

View your accounts

The **Organization** page lists all OUs and accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can view and enroll member accounts into AWS Control Tower—individually or by OU groups—if each account meets the prerequisites for enrollment.

To view a specific account on the **Organization** page, you can choose **Accounts only** from the dropdown menu at the upper right, and then select the name of your account from the table. Alternatively, you can select the name of the parent OU from the table, and you can view a list of all accounts within that OU on the **Details** page for that OU.

On the **Organization** page and the **Account details** page, you can see the account's **State**, which is one of these:

- **Not enrolled –** The account is a member of the parent OU, but it is not fully managed by AWS Control Tower. If the parent OU is registered, the account is governed by the preventive controls configured for its registered parent OU, but the OU's detective controls do not apply to this account. If the parent OU is unregistered, no controls apply to this account.
- **Enrolling –** The account is being brought into governance by AWS Control Tower. We are aligning the account with the control configuration for the parent OU. This process may require several minutes per account resource.
- **Enrolled –** The account is governed by the controls configured for its parent OU. It is fully managed by AWS Control Tower.

- **Enrollment failed –** The account could not be enrolled in AWS Control Tower. For more information, see Common causes for failure of enrollment.
- **Update available –** The account has an update available. Accounts in this state are still **Enrolled**, but the account must be updated to reflect recent changes made to your environment. To update a single account, navigate to the account detail page and select **Update account**.

  If you have multiple accounts with this state under a single OU, you can choose to **Re-register** the OU and update those accounts together.

Resources created in the shared accounts

This section shows the resources that AWS Control Tower creates in the shared accounts, when you set up your landing zone.

For information about member account resources, see Resource Considerations for Account Factory.

Management account resources

When you set up your landing zone, the following AWS resources are created within your management account.

AWS service	Resource type	Resource name
AWS Organizations	Accounts	audit log archive
AWS Organizations	OUs	Security Sandbox
AWS Organizations	Service Control Policies	aws-guardrails-*
AWS CloudFormation	Stacks	AWSControlTowerBP-BASELINE-CLOUDTRAIL-MASTER

AWS service	Resource type	Resource name
		AWSControlTowerBP-BASELINE-CONFIG-MASTER (in version 2.6 and later)

AWS service	Resource type	Resource name
AWS CloudFormation	StackSets	AWSControlTowerBP-BASELINE-CLOUDTRAIL (Not deployed in 3.0 and later) AWSControlTowerBP_BASELINE_SERVICE_LINKED_ROLE (Deployed in 3.2 and later) AWSControlTowerBP-BASELINE-CLOUDWATCH AWSControlTowerBP-BASELINE-CONFIG AWSControlTowerBP-BASELINE-ROLES AWSControlTowerBP-BASELINE-SERVICE-ROLES AWSControlTowerBP-SECURITY-TOPICS AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED AWSControlTowerLoggingResources

AWS service	Resource type	Resource name
		AWSControlTowerSecurityResources AWSControlTowerExecutionRole
AWS Service Catalog	Product	AWS Control Tower Account Factory
AWS Config	Aggregator	aws-controltower-ConfigAggregatorForOrganizations
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs
AWS Identity and Access Management	Roles	AWSControlTowerAdmin AWSControlTowerStackSetRole AWSControlTowerCloudTrailRolePolicy
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy AWSControlTowerAdminPolicy AWSControlTowerCloudTrailRolePolicy AWSControlTowerStackSetRolePolicy

AWS service	Resource type	Resource name
AWS IAM Identity Center	Directory groups	AWSAccountFactory AWSAuditAccountAdmins AWSControlTowerAdmins AWSLogArchiveAdmins AWSLogArchiveViewers AWSSecurityAuditors AWSSecurityAuditPowerUsers AWSServiceCatalogAdmins
AWS IAM Identity Center	Permission Sets	AWSAdministratorAccess AWSPowerUserAccess AWSServiceCatalogA dminFullAccess AWSServiceCatalogE ndUserAccess AWSReadOnlyAccess AWSOrganizationsFullAccess

Note

The AWS CloudFormation StackSet `BP_BASELINE_CLOUDTRAIL` is not deployed in landing zone versions 3.0 or later. However, it continues to exist in earlier versions of the landing zone, until you update your landing zone.

Log archive account resources

When you set up your landing zone, the following AWS resources are created within your log archive account.

AWS service	Resource type	Resource Name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED- StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH- StackSet-AWSControlTowerBP-BASELINE-CONFIG- StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL- StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES- StackSet-AWSControlTowerBP-BASELINE-SERVICE-LINKED-ROLE-(In 3.2 and later) StackSet-AWSControlTowerBP-BASELINE-ROLES-

AWS service	Resource type	Resource Name
		StackSet-AWSControlTowerLoggingResources-
AWS Config	AWS Config Rules	AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITED AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBIT
AWS CloudTrail	Trails	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	/aws/lambda/aws-controltower-NotificationForwarder
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole AWSControlTowerExecution

AWS service	Resource type	Resource Name
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy
Amazon Simple Notification Service	Topics	aws-controltower-SecurityNotifications
AWS Lambda	Applications	StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*
AWS Lambda	Functions	aws-controltower-NotificationForwarder
Amazon Simple Storage Service	Buckets	aws-controltower-logs-* aws-controltower-s3-access-logs-*

Audit account resources

When you set up your landing zone, the following AWS resources are created within your audit account.

AWS service	Resource type	Resource name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED- StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED-

AWS service	Resource type	Resource name
		StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH- StackSet-AWSControlTowerBP-BASELINE-CONFIG- StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL- StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES- StackSet-AWSControlTowerBP-BASELINE-SERVICE-LINKED-ROLE-(In 3.2 and later) StackSet-AWSControlTowerBP-SECURITY-TOPICS- StackSet-AWSControlTowerBP-BASELINE-ROLES- StackSet-AWSControlTowerSecurityResources-*
AWS Config	Aggregator	aws-controltower-GuardrailsComplianceAggregator

AWS service	Resource type	Resource name
AWS Config	AWS Config Rules	AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITED AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBITED
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	/aws/lambda/aws-controltower-NotificationForwarder

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole aws-controltower-AuditAdministratorRole aws-controltower-AuditReadOnlyRole AWSControlTowerExecution
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy
Amazon Simple Notification Service	Topics	aws-controltower-AggregateSecurityNotifications aws-controltower-AllConfigNotifications aws-controltower-SecurityNotifications
AWS Lambda	Functions	aws-controltower-NotificationForwarder

About the shared accounts

Three special AWS accounts are associated with AWS Control Tower; the management account, the **audit** account, and the **log archive** account. These accounts usually are referred to as *shared accounts*, or sometimes as *core accounts*.

- You can select customized names for the audit and log archive accounts when you're setting up your landing zone. For information about changing an account name, see Externally changing AWS Control Tower resource names.
- You also can specify an existing AWS account as an AWS Control Tower security or logging account, during the initial landing zone setup process. This option eliminates the need for AWS Control Tower to create new, shared accounts. (This is a one-time selection.)

For more information about the shared accounts and their associated resources, see Resources created in the shared accounts.

Management account

This AWS account launches AWS Control Tower. By default, the root user for this account and the IAM user or IAM administrator user for this account have full access to all resources within your landing zone.

> **Note**
>
> As a best practice, we recommend signing in as an IAM Identity Center user with **Administrator** privileges when performing administrative functions within the AWS Control Tower console, instead of the signing in as the root user or IAM administrator user for this account.

For more information about the roles and resources available in the management account, see Resources created in the shared accounts.

Log archive account

The log archive shared account is set up automatically when you create your landing zone.

This account contains a central Amazon S3 bucket for storing a copy of all AWS CloudTrail and AWS Config log files for all other accounts in your landing zone. As a best practice, we recommend restricting log archive account access to teams responsible for compliance and investigations, and their related security or audit tools. This account can be used for automated security audits, or to host custom AWS Config Rules, such as Lambda functions, to perform remediation actions.

> **Amazon S3 bucket policy**
>
> For AWS Control Tower landing zone version 3.3 and later, accounts must meet an `aws:SourceOrgID` condition for any write permissions to your Audit bucket. This condition ensures that CloudTrail only can write logs on behalf of accounts within your organization to your S3 bucket; it prevents CloudTrail logs outside your organization from writing to your AWS Control Tower S3 bucket. For more information, see AWS Control Tower landing zone version 3.3.

For more information about the roles and resources available in the log archive account, see Log archive account resources

> **Note**
>
> These logs cannot be changed. All logs are stored for the purposes of audit and compliance investigations related to account activity.

Audit account

This shared account is set up automatically when you create your landing zone.

The audit account should be restricted to security and compliance teams with auditor (read-only) and administrator (full-access) cross-account roles to all accounts in the landing zone. These roles are intended to be used by security and compliance teams to:

- Perform audits through AWS mechanisms, such as hosting custom AWS Config rule Lambda functions.
- Perform automated security operations, such as remediation actions.

The audit account also receives notifications through the Amazon Simple Notification Service (Amazon SNS) service. Three categories of notification can be received:

- **All Configuration Events –** This topic aggregates all CloudTrail and AWS Config notifications from all accounts in your landing zone.
- **Aggregate Security Notifications –** This topic aggregates all security notifications from specific CloudWatch events, AWS Config Rules compliance status change events, and GuardDuty findings.
- **Drift Notifications –** This topic aggregates all the drift warnings discovered across all accounts, users, OUs, and SCPs in your landing zone. For more information on drift, see Detect and resolve drift in AWS Control Tower.

Audit notifications that are triggered within a member account also can send alerts to a local Amazon SNS topic. This functionality allows account administrators to subscribe to audit notifications that are specific to an individual member account. As a result, administrators can resolve issues that affect an individual account, while still aggregating all account notifications to your centralized audit account. For more information, see Amazon Simple Notification Service Developer Guide.

For more information about the roles and resources available in the audit account, see Audit account resources.

For more information about programmatic auditing, see Programmatic roles and trust relationships for the AWS Control Tower audit account.

> **Important**
>
> The email address you provide for the audit account receives **AWS Notification - Subscription Confirmation** emails from every AWS Region supported by AWS Control Tower. To receive compliance emails in your audit account, you must choose the **Confirm subscription** link within each email from each AWS Region supported by AWS Control Tower.

About member accounts

Member accounts are the accounts through which your users perform their AWS workloads. These member accounts can be created in Account Factory, by IAM Identity Center users with **Admin** privileges in the Service Catalog console, or by automated methods. When created, these member accounts exist in an OU that was created in the AWS Control Tower console, or registered with AWS Control Tower. For more information, see these related topics:

- Provision and manage accounts with Account Factory
- Automate tasks in AWS Control Tower
- AWS Organizations Terminology and Concepts in the *AWS Organizations User Guide*.

Also see Provision accounts with AWS Control Tower Account Factory for Terraform (AFT) .

> **Accounts and controls**
>
> Member accounts can be *enrolled* in AWS Control Tower, or they can be *unenrolled*. Controls apply differently to enrolled and unenrolled accounts, and controls may apply to accounts in nested OUs based on inheritance.

For information about member account resources that AWS Control Tower allocates, see Resource Considerations for Account Factory.

Enroll an existing AWS account

You can extend AWS Control Tower governance to an individual, existing AWS account when you *enroll* it into an organizational unit (OU) that's already governed by AWS Control Tower. Eligible accounts exist in *unregistered OUs that are part of the same AWS Organizations organization* as the AWS Control Tower OU.

> **Note**
>
> You cannot enroll an existing account to serve as your audit or log archive account except during initial landing zone setup.

Set up trusted access first

Before you can enroll an existing AWS account into AWS Control Tower you must give permission for AWS Control Tower to manage, or *govern*, the account. Specifically, AWS Control Tower requires permission to establish trusted access between AWS CloudFormation and AWS Organizations on your behalf, so that AWS CloudFormation can deploy your stack automatically to the accounts in your selected organization. With this trusted access, the `AWSControlTowerExecution` role conducts activities required to manage each account. That's why you must add this role to each account before you enroll it.

When trusted access is enabled, AWS CloudFormation can create, update, or delete stacks across multiple accounts and AWS Regions with a single operation. AWS Control Tower relies on this trust capability so it can apply roles and permissions to existing accounts before it moves them into a registered organizational unit, and thereby brings them under governance.

To learn more about trusted access and AWS CloudFormation StackSets, see AWS CloudFormationStackSets and AWS Organizations.

What happens during account enrollment

During the enrollment process, AWS Control Tower performs these actions:

- Baselines the account, which includes deploying these stack sets:
  - `AWSControlTowerBP-BASELINE-CLOUDTRAIL`
  - `AWSControlTowerBP-BASELINE-CLOUDWATCH`
  - `AWSControlTowerBP-BASELINE-CONFIG`
  - `AWSControlTowerBP-BASELINE-ROLES`
  - `AWSControlTowerBP-BASELINE-SERVICE-ROLES`
  - `AWSControlTowerBP-BASELINE-SERVICE-LINKED-ROLES`
  - `AWSControlTowerBP-VPC-ACCOUNT-FACTORY-V1`

  It is a good idea to review the templates of these stack sets and make sure that they don’t conflict with your existing policies.
- Identifies the account through AWS IAM Identity Center or AWS Organizations.
- Places the account into the OU that you've specified. Be sure to apply all SCPs that are applied in the current OU, so that your security posture remains consistent.
- Applies mandatory controls to the account by means of the SCPs that apply to the selected OU as a whole.
- Enables AWS Config and configures it to record all resources in the account.
- Adds the AWS Config rules that apply the AWS Control Tower detective controls to the account.

Accounts and organization-level CloudTrail trails

All member accounts in an OU are governed by the AWS CloudTrail trail for the OU, enrolled or not:

- When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the new organization. If you have an existing deployment of a CloudTrail trail, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower.
- If you move an account into a registered OU—for example by means of the AWS Organizations console—and you do not proceed to enroll the account into AWS Control Tower, you may wish to remove any remaining account-level trails for the account. If you have an existing deployment of a CloudTrail trail, you will incur duplicate CloudTrail charges.

If you update your landing zone and choose to opt out of organization-level trails, or if your landing zone is older than version 3.0, organization-level CloudTrail trails do not apply to your accounts.

Enrolling existing accounts with VPCs

AWS Control Tower handles VPCs differently when you provision a new account in Account Factory than when you enroll an existing account.

- When you create a new account, AWS Control Tower automatically removes the AWS default VPC and creates a new VPC for that account.
- When you enroll an existing account, AWS Control Tower does not create a new VPC for that account.
- When you enroll an existing account, AWS Control Tower does not remove any existing VPC or AWS default VPC associated with the account.

Tip

You can change the default behavior for new accounts by configuring Account Factory, so it does not set up a VPC by default for accounts in your organization under AWS Control Tower. For more information, see [Create an Account in AWS Control Tower Without a VPC](#).

Prerequisites for enrollment

These prerequisites are required before you can enroll an existing AWS account in AWS Control Tower:

1. To enroll an existing AWS account, the `AWSControlTowerExecution` role must be present in the account you are enrolling. You can review Enroll an account for details and instructions.
2. In addition to the `AWSControlTowerExecution` role, the existing AWS account you want to enroll must have the following permissions and trust relationships in place. Otherwise, enrollment will fail.

   Role Permission: `AdministratorAccess` (AWS managed policy)

   Role Trust Relationship:

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Principal": {
                   "AWS": "arn:aws:iam::111122223333:root"
               },
               "Action": "sts:AssumeRole"
           }
       ]
   }
   ```

3. We recommend that the account should not have an AWS Config configuration recorder or delivery channel. These may be deleted or modified through the AWS CLI before you can enroll an account. Otherwise, review Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.
4. The account that you wish to enroll must exist in the same AWS Organizations organization as the AWS Control Tower management account. The account that exists can be enrolled *only* into the same organization as the AWS Control Tower management account, in an OU that already is registered with AWS Control Tower.

To check other prerequisites for enrollment, see Getting Started with AWS Control Tower.

Note

When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the AWS Control Tower organization. If you have an existing deployment of a CloudTrail trail, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower.

Enroll an existing account

The **Enroll account** feature is available in the AWS Control Tower console, for enrolling existing AWS accounts so that they are governed by AWS Control Tower. For more information, see Enroll an existing AWS account.

The **Enroll account** capability is available when your landing zone is not in a state of drift.

To view your eligible accounts in the console:

- Navigate to the **Organization** page in AWS Control Tower.
- Find the name of the account you wish to enroll. To find it, choose **Accounts only** from the dropdown menu at the upper right, and then locate the account name in the filtered table.
- Follow the steps for enrolling an individual account, as shown in the Steps to enroll an account manually section.

Note

When you are enrolling an existing AWS account, be sure to verify the existing email address. Otherwise, a new account may be created.

Certain errors may require that you refresh the page and try again. If your landing zone is in a state of drift, you may not be able to use the **Enroll account** capability successfully. You'll need to provision new accounts through Account Factory until your landing zone drift has been resolved.

When you enroll accounts from the AWS Control Tower console, you must be signed into an account with a user that has the `AWSServiceCatalogEndUserFullAccess` policy enabled, along with Administrator access permissions to use the AWS Control Tower console, and you cannot be signed in as the root user.

Accounts that you enroll may be updated by means of the AWS Control Tower account factory, as you would update any other account. Update procedures are given in the section called Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.

Steps to enroll an account manually

After the **AdministratorAccess** permission (policy) is in place in your existing account, follow these steps to enroll the account:

To enroll an individual account in AWS Control Tower from the console

- Navigate to the AWS Control Tower **Organization** page.
- On the **Organization** page, accounts that are eligible to be enrolled allow you to select **Enroll** from the **Actions** dropdown menu at the top of the section. These accounts also show an **Enroll account** button when you view them on the **Account details** page.
- When you choose **Enroll account**, you'll see an **Enroll account** page, where you are prompted to add the `AWSControlTowerExecution` role to the account. For some instructions, see Manually add the required IAM role to an existing AWS account and enroll it.
- Next, select a registered OU from the drop down list. If the account is already in a registered OU, this list will show the OU.
- Choose **Enroll account**.
- You'll see a modal reminder to add the `AWSControlTowerExecution` role and confirm the action.
- Choose **Enroll**.
- AWS Control Tower begins the process of enrollment, and you are directed back to the **Account details** page.

Common causes for failure of enrollment

- To enroll an existing account, the `AWSControlTowerExecution` role must be present in the account you're enrolling.
- Your IAM principal may lack the necessary permissions to provision an account.
- AWS Security Token Service (AWS STS) is disabled in your AWS account in your home Region, or in any Region supported by AWS Control Tower.
- You may be signed in to an account that needs to be added to the Account Factory Portfolio in AWS Service Catalog. The account must be added before you'll have access to Account Factory

so you can create or enroll an account in AWS Control Tower. If the appropriate user or role is not added to the Account Factory portfolio, you'll receive an error when you attempt to add an account. For instructions on how to grant access to AWS Service Catalog portfolios, see Granting access to users.

- You may be signed in as root.
- The account you're trying to enroll may have AWS Config settings that are residual. In particular, the account may have a configuration recorder or delivery channel. These must be deleted or modified through the AWS CLI before you can enroll an account. For more information, see Enroll accounts that have existing AWS Config resources and Interact with AWS Control Tower through AWS CloudShell.
- If the account belongs to another OU with a management account, including another AWS Control Tower OU, you must terminate the account in its current OU before it can join another OU. Existing resources must be removed in the original OU. Otherwise, enrollment will fail.
- Account provisioning and enrollment fails if your destination OU's SCPs don't allow you to create all of the resources required for that account. For example, an SCP in your destination OU may block resource creation without certain tags. In this case, account provisioning or enrollment fails, because AWS Control Tower does not support tagging of resources. For help, contact your account representative, or Support.

For more information about how AWS Control Tower works with roles when you're creating new accounts or enrolling existing accounts, see Roles and accounts.

> **Tip**
>
> If you cannot confirm that an existing AWS account meets the enrollment prerequisites, you can set up an **Enrollment OU** and enroll the account into that OU. After enrollment is successful, you can move the account to the desired OU. If enrollment happens to fail, no other accounts or OUs are affected by the failure.

If you have doubts that your existing accounts and their configurations are compatible with AWS Control Tower, you can follow the best practice recommended in the following section.

Recommended: You can set up a two-step approach to account enrollment

- First, use an AWS Config *conformance pack* to evaluate how your accounts may be affected by some AWS Control Tower controls. To determine how enrollment into AWS Control Tower may affect your accounts, see [Extend AWS Control Tower governance using AWS Config conformance packs].

- Next, you may wish to enroll the account. If the compliance results are satisfactory, the migration path is easier because you can enroll the account without unexpected consequences.
- After you've done your evaluation, if you decide to set up an AWS Control Tower landing zone, you may need to remove the AWS Config delivery channel and configuration recorder that were created for your evaluation. Then you'll be able to set up AWS Control Tower successfully.

> **Note**
>
> The conformance pack also works in situations where the accounts are located in OUs registered by AWS Control Tower, but the workloads run within AWS Regions that don't have AWS Control Tower support. You can use the conformance pack to manage resources in accounts that exist in Regions where AWS Control Tower is not deployed.

If the account does not meet the prerequisites

Remember that, as a prerequisite, accounts eligible to be enrolled into AWS Control Tower governance must be part of the same overall organization. To fulfill this prerequisite for account enrollment, you can follow these preparatory steps to move an account into the same organization as AWS Control Tower.

Preparatory steps to bring an account into the same organization as AWS Control Tower

1. Drop the account from its existing organization. You must provide a separate payment method if you use this approach.
2. Invite the account to join the AWS Control Tower organization. For more information, see [Inviting an AWS account to join your organization] in the *AWS Organizations User Guide*.
3. Accept the invitation. The account shows up in the root of the organization. This step moves the account into the same organization as AWS Control Tower. and establishes SCPs and consolidated billing.

> **Tip**
>
> You can send the invitation for the new organization before the account drops out of the old organization. The invitation will be waiting when the account officially drops out of its existing organization.

Steps to fulfill the remaining prerequisites:

1. Create the necessary `AWSControlTowerExecution` role.
2. Clear out the default VPC. (This part is optional. AWS Control Tower doesn't change your existing default VPC.)
3. Delete or modify any existing AWS Config configuration recorder or delivery channel through the AWS CLI or AWS CloudShell. For more information, see Example AWS Config CLI commands for resource status and Enroll accounts that have existing AWS Config resources

After you've completed these preparatory steps, you can enroll the account into AWS Control Tower. For more information, see Steps to enroll an account manually. This step brings the account into full AWS Control Tower governance.

Optional steps to deprovision an account, so it can be enrolled and keep its stack

1. To keep the applied AWS CloudFormation stack, delete the stack instance from the stack sets, and choose **Retain stacks** for the instance.
2. Terminate the account provisioned product in AWS Service Catalog Account Factory. (This step only removes the provisioned product from AWS Control Tower. It doesn't delete the account.)
3. Set up the account with the necessary billing details, as required for any account that doesn't belong to an organization. Then remove the account from the organization. (You do this, so the account doesn't count against the total in your AWS Organizations quota.)
4. Clean up the account if resources remain, and then close it, following the account closure steps in Unenroll an account.
5. If you have a **Suspended** OU with defined controls, you can move the account there instead of doing Step 1.

Example AWS Config CLI commands for resource status

Here are some example AWS Config CLI commands you can use to determine the status of your configuration recorder and delivery channel.

View commands:

- `aws configservice describe-delivery-channels`
- `aws configservice describe-delivery-channel-status`
- `aws configservice describe-configuration-recorders`

The normal response is something like `"name": "default"`

Delete commands:

- `aws configservice stop-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
- `aws configservice delete-delivery-channel --delivery-channel-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
- `aws configservice delete-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*

Manually add the required IAM role to an existing AWS account and enroll it

If you've already set up your AWS Control Tower landing zone, you can begin enrolling your organization's accounts into an OU that is registered with AWS Control Tower. If you haven't set up your landing zone, follow the steps as described in the *AWS Control Tower User Guide* at Getting Started, Step 2. After the landing zone is ready, complete the following steps to bring existing accounts into governance by AWS Control Tower, manually.

Be sure to review the Prerequisites for enrollment noted previously in this chapter.

Before enrolling an account with AWS Control Tower, you must give AWS Control Tower permission to manage that account. To do so, you'll add a role that has full access to the account, as shown in the steps that follow. These steps must be performed for each account that you enroll.

For each account:

Step 1: Sign in with administrator access to the management account of the organization that currently contains the account you wish to enroll.

For example, if you created this account from AWS Organizations and you use a cross-account IAM role to sign in, then you may follow these steps:

1. Sign in to your organization’s management account.
2. Go to **AWS Organizations**.
3. Under **Accounts**, select the account you want to enroll and copy its account ID.
4. Open the account dropdown menu on the top navigation bar and choose **Switch Role**.
5. On the **Switch role** form, fill in the following fields:
   - Under **Account**, enter the account ID you copied.
   - Under **Role**, enter the name of the IAM role that enables cross-account access to this account. The name of this role was defined when the account was created. If you did not specify a role name when you created the account, enter the default role name, `OrganizationAccountAccessRole`.
6. Choose **Switch Role**.
7. You should now be signed into the AWS Management Console as the child account.
8. When you’re finished, stay in the child account for the next part of the procedure.
9. Make note of the management account ID, because you will need to enter it in the next step.

Step 2: Give AWS Control Tower permission to manage the account.

1. Go to **IAM**.
2. Go to **Roles**.
3. Choose **Create role**.
4. When asked to select which service the role is for, choose **Custom trust policy**.
5. Copy the code example shown here and paste it into the Policy Document. Replace the string *`Management Account ID`* with the actual management account ID of your management account. Here is the policy to paste:

   JSON

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

6. When asked to attach policies, choose **AdministratorAccess**.
7. Choose **Next:Tags**.
8. You may see an optional screen titled **Add tags**. Skip this screen for now by choosing **Next:Review**
9. On the **Review** screen, in the **Role name** field, enter `AWSControlTowerExecution`.
10. Enter a brief description in the **Description** box, such as *Allows full account access for enrollment.*
11. Choose **Create role**.

Step 3: Enroll the account by moving it into a registered OU, and verify enrollment.

After you've set up the necessary permissions by creating the role, follow these steps to enroll the account and verify enrollment.

1. **Sign in again as Admin and go to AWS Control Tower.**
2. **Enroll the account.**
   - From the **Organization** page in AWS Control Tower, select your account, then choose **Enroll** from the **Actions** dropdown menu at the upper right.
   - Follow the steps for enrolling an individual account, as shown on the [Steps to enroll an account manually](#) page.
3. **Verify enrollment.**
   - From AWS Control Tower, choose **Organization** in the left navigation.

- Look for the account you have recently enrolled. Its initial state will show a status of **Enrolling**.
- When the state changes to **Enrolled**, the move was successful.

To continue this process, sign into each account in your organization that you want to enroll in AWS Control Tower. Repeat the prerequisite steps and the enrollment steps for each account.

The following YAML template may assist you in creating the required role in an account, so that it can be enrolled programmatically.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure the AWSControlTowerExecution role to enable use of your
  account as a target account in AWS CloudFormation StackSets.
Parameters:
  AdministratorAccountId:
    Type: String
    Description: AWS Account Id of the administrator account (the account in which
      StackSets will be created).
    MaxLength: 12
    MinLength: 12
Resources:
  ExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: AWSControlTowerExecution
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              AWS:
                - !Ref AdministratorAccountId
            Action:
              - sts:AssumeRole
      Path: /
      ManagedPolicyArns:
        - !Sub arn:${AWS::Partition}:iam::aws:policy/AdministratorAccess
```

Enroll accounts that have existing AWS Config resources

This topic provides a step-by-step approach for how to enroll accounts that have existing AWS Config resources. For examples of how to check your existing resources, see Example AWS Config CLI commands for resource status.

> **Note**
>
> If you plan to bring existing AWS accounts into AWS Control Tower as **Audit** and **Log archive** accounts, and if those accounts have existing AWS Config resources, you must delete the existing AWS Config resources completely, before you can enroll these accounts into AWS Control Tower for this purpose. For accounts that are not intended to become **Audit** and **Log archive** accounts, you can modify the existing Config resources.

Examples of AWS Config resources

Here are some types of AWS Config resources that your account could have already. These resources may need to be modified so that you can enroll your account into AWS Control Tower.

- AWS Config recorder
- AWS Config delivery channel
- AWS Config aggregation authorization

Assumptions

- You have deployed an AWS Control Tower landing zone
- Your account is not enrolled with AWS Control Tower already.
- Your account has at least one pre-existing AWS Config resource in at least one of the AWS Control Tower Regions governed by the management account.
- Your account is not the AWS Control Tower management account.
- Your account is not in governance drift.

For a blog that describes an automated approach to enrolling accounts with existing AWS Config resources, see Automate enrollment of accounts with existing AWS Config resources into AWS

Control Tower. You'll be able to submit a single support ticket for all of the accounts you wish to enroll, as described in Step 1: Contact customer support with a ticket, to add the account to the AWS Control Tower allow list, which follows.

Limitations

- The account can be enrolled only by using the AWS Control Tower workflow for extending governance.
- If the resources are modified and create drift on the account, AWS Control Tower does not update the resources.
- AWS Config resources in Regions that are not governed by AWS Control Tower are not changed.

> **Note**
>
> If you attempt to enroll an account that has existing Config resources, without having the account added to the allow list, enrollment will fail. Thereafter, if you subsequently try to add that same account to the allow list, AWS Control Tower cannot validate that the account is provisioned correctly. You must deprovision the account from AWS Control Tower before you can request the allow list and then enroll it. If you only move the account to a different AWS Control Tower OU, it causes governance drift, which also prevents the account from being added to the allow list.

This process has 5 main steps.

1. Add the account to the AWS Control Tower allow list.
2. Create a new IAM role in the account.
3. Modify pre-existing AWS Config resources.
4. Create AWS Config resources in AWS Regions where they don't exist.
5. Enroll the account with AWS Control Tower.

Before you proceed, consider the following expectations regarding this process.

- AWS Control Tower does not create any AWS Config resources in this account.
- After enrollment, AWS Control Tower controls automatically protect the AWS Config resources you created, including the new IAM role.

- If any changes are made to the AWS Config resources after enrollment, those resources must be updated to align with AWS Control Tower settings before you can re-enroll the account.

Step 1: Contact customer support with a ticket, to add the account to the AWS Control Tower allow list

Include this phrase in your ticket subject line:

Enroll accounts that have existing AWS Config resources into AWS Control Tower

Include the following details in the body of your ticket:

- Management account number
- Account numbers of member accounts that have existing AWS Config resources
- Your selected home Region for AWS Control Tower setup

> **Note**
>
> The required time for adding your account to the allow list is 2 business days.

Step 2: Create a new IAM role in the member account

1. Open the AWS CloudFormation console for the member account.
2. Create a new stack using the following template

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config

Resources:
  CustomerCreatedConfigRecorderRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: aws-controltower-ConfigRecorderRole-customer-created
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
```

```
            Principal:
              Service:
                - config.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWS_ConfigRole
        - arn:aws:iam::aws:policy/ReadOnlyAccess
```

3. Provide the name for the stack as **CustomerCreatedConfigRecorderRoleForControlTower**
4. Create the stack.

> **Note**
>
> Any SCPs that you create should exclude an `aws-controltower-ConfigRecorderRole*` role. Do not modify the permissions that restrict the ability for AWS Config rules to perform evaluations.
> Follow these guidelines so that you don't receive an `AccessDeniedException` when you have SCPs that block `aws-controltower-ConfigRecorderRole*` from calling Config.

Step 3: Identify the AWS Regions with pre-existing resources

For each governed Region (AWS Control Tower governed) in the account, identify and note the Regions that have at least one of the existing AWS Config resource example types shown previously.

Step 4: Identify the AWS Regions without any AWS Config resources

For each governed Region (AWS Control Tower governed) in the account, identify and note the Regions in which there are no AWS Config resources of the example types shown previously.

Step 5: Modify the existing resources in each AWS Region

For this step, the following information is needed about your AWS Control Tower setup.

- `LOGGING_ACCOUNT` - the Logging account ID
- `AUDIT_ACCOUNT` - the Audit account ID

- `IAM_ROLE_ARN` - the IAM role ARN created in Step 1
- `ORGANIZATION_ID` - the organization ID for the management account
- `MEMBER_ACCOUNT_NUMBER` - the member account that is being modified
- `HOME_REGION` - the home Region for AWS Control Tower setup.

Modify each existing resource by following the instructions given in sections 5a through 5c, which follow.

Step 5a. AWS Config recorder resources

Only one AWS Config recorder can exist per AWS Region. If one exists, modify the settings as shown. Replace the item `GLOBAL_RESOURCE_RECORDING` with **true** in your home Region. Replace the item with **false** for other Regions where an AWS Config recorder exists.

- **Name:** DON'T CHANGE
- **RoleARN:** `IAM_ROLE_ARN`
  - **RecordingGroup:**
  - **AllSupported:** true
  - **IncludeGlobalResourceTypes:** `GLOBAL_RESOURCE_RECORDING`
  - **ResourceTypes:** Empty

This modification can be made through the AWS CLI using the following command. Replace the string `RECORDER_NAME` with the existing AWS Config recorder name.

```
aws configservice put-configuration-recorder --configuration-recorder
  name=RECORDER_NAME,roleARN=arn:aws:iam::MEMBER_ACCOUNT_NUMBER:role/
aws-controltower-ConfigRecorderRole-customer-created --recording-group
 allSupported=true,includeGlobalResourceTypes=GLOBAL_RESOURCE_RECORDING --
region CURRENT_REGION
```

Step 5b. Modify AWS Config delivery channel resources

Only one AWS Config delivery channel can exist per Region. If another exists, modify the settings as shown.

- **Name:** DON'T CHANGE
- **ConfigSnapshotDeliveryProperties:** TwentyFour_Hours
- **S3BucketName:** The logging bucket name from the AWS Control Tower logging account

  `aws-controltower-logs-LOGGING_ACCOUNT-HOME_REGION`
- **S3KeyPrefix:** *ORGANIZATION_ID*
- **SnsTopicARN:** The SNS topic ARN from the audit account, with the following format:

  `arn:aws:sns:CURRENT_REGION:AUDIT_ACCOUNT:aws-controltower-AllConfigNotifications`

This modification can be made through the AWS CLI using the following command. Replace the string *DELIVERY_CHANNEL_NAME* with the existing AWS Config recorder name.

```
aws configservice put-delivery-channel --delivery-channel
 name=DELIVERY_CHANNEL_NAME,s3BucketName=aws-controltower-
logs-LOGGING_ACCOUNT_ID-
HOME_REGION,s3KeyPrefix="ORGANIZATION_ID",configSnapshotDeliveryProperties={deliveryFrequency=T
controltower-AllConfigNotifications --region CURRENT_REGION
```

Step 5c. Modify AWS Config aggregation authorization resources

Multiple aggregation authorizations can exist per Region. AWS Control Tower requires an aggregation authorization that specifies the audit account as the authorized account, and has the home Region for AWS Control Tower as the authorized Region. If it doesn't exist, create a new one with the following settings:

- **AuthorizedAccountId:** The Audit account ID
- **AuthorizedAwsRegion:** The home Region for the AWS Control Tower setup

This modification can be made through the AWS CLI using the following command:

```
aws configservice put-aggregation-authorization --authorized-account-
id AUDIT_ACCOUNT_ID --authorized-aws-region HOME_REGION --region
CURRENT_REGION
```

Step 6: Create resources where they don’t exist, in Regions governed by AWS Control Tower

Revise the AWS CloudFormation template, so that in your home Region the **IncludeGlobalResourcesTypes** parameter has the value `GLOBAL_RESOURCE_RECORDING`, as shown in the example that follows. Also update the required fields in the template, as specified in this section.

Replace the item `GLOBAL_RESOURCE_RECORDING` with **true** in your home Region. Replace the item with **false** for other Regions where an AWS Config recorder does not exist.

1. Navigate to the management account’s AWS CloudFormation console.
2. Create a new StackSet with the name **CustomerCreatedConfigResourcesForControlTower**.
3. Copy and update the following template:

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config
Resources:
  CustomerCreatedConfigRecorder:
    Type: AWS::Config::ConfigurationRecorder
    Properties:
      Name: aws-controltower-BaselineConfigRecorder-customer-created
      RoleARN: !Sub arn:aws:iam::${AWS::AccountId}:role/aws-controltower-
ConfigRecorderRole-customer-created
      RecordingGroup:
        AllSupported: true
        IncludeGlobalResourceTypes: GLOBAL_RESOURCE_RECORDING
        ResourceTypes: []
  CustomerCreatedConfigDeliveryChannel:
    Type: AWS::Config::DeliveryChannel
    Properties:
      Name: aws-controltower-BaselineConfigDeliveryChannel-customer-created
      ConfigSnapshotDeliveryProperties:
        DeliveryFrequency: TwentyFour_Hours
      S3BucketName: aws-controltower-logs-LOGGING_ACCOUNT-HOME_REGION
      S3KeyPrefix: ORGANIZATION_ID
      SnsTopicARN: !Sub arn:aws:sns:${AWS::Region}:AUDIT_ACCOUNT:aws-controltower-
AllConfigNotifications
  CustomerCreatedAggregationAuthorization:
    Type: "AWS::Config::AggregationAuthorization"
    Properties:
```

```
        AuthorizedAccountId: AUDIT_ACCOUNT
        AuthorizedAwsRegion: HOME_REGION
```

Update the template with required fields:

a. In the **S3BucketName** field, replace the *LOGGING_ACCOUNT_ID* and *HOME_REGION*

b. In the **S3KeyPrefix** field, replace the *ORGANIZATION_ID*

c. In the **SnsTopicARN** field, replace the *AUDIT_ACCOUNT*

d. In the **AuthorizedAccountId** field, replace the *AUDIT_ACCOUNT*

e. In the **AuthorizedAwsRegion** field, replace the *HOME_REGION*

4. During deployment on the AWS CloudFormation console, add the member account number.
5. Add the AWS Regions that were identified in Step 4.
6. Deploy the stack set.

Step 7: Register the OU with AWS Control Tower

In the AWS Control Tower dashboard, register the OU.

> **Note**
>
> The **Enroll account** workflow will not succeed for this task. You must choose **Register OU** or **Re-register OU**.

Provision and manage accounts with Account Factory

This chapter includes an overview and procedures for provisioning new member accounts in an AWS Control Tower landing zone with Account Factory.

Permissions for configuring and provisioning accounts

The AWS Control Tower Account Factory enables cloud administrators and users in AWS IAM Identity Center to provision accounts in your landing zone. By default, IAM Identity Center users that provision accounts must be in the `AWSAccountFactory` group or the management group.

> **Note**
>
> Exercise caution when working from the management account, as you would when using any account that has permissions across your organization.

The AWS Control Tower management account has a trust relationship with the `AWSControlTowerExecution` role, which allows account setup from the management account, including some automated account setup. For more information about the `AWSControlTowerExecution` role, see Roles and accounts.

> **Note**
>
> To enroll an existing AWS account into AWS Control Tower, that account must have the `AWSControlTowerExecution` role enabled. For more information about how to enroll an existing account, see Enroll an existing AWS account.

For more information about permissions, see Permissions required for accounts.

Provision accounts with AWS Service Catalog Account Factory

The following procedure describes how to create and provision accounts as a user in IAM Identity Center through AWS Service Catalog. This procedure also is referred to as *advanced account provisioning*, or *manual account provisioning*. Optionally, you may be able to provision accounts programmatically, with the AWS CLI or with AWS Control Tower Account Factory for Terraform (AFT). You may be able to provision customized accounts in the console if you've previously set up custom blueprints. For more information about customization, see Customize accounts with Account Factory Customization (AFC).

To provision accounts individually in Account Factory, as a user

1. Sign in from your user portal URL.
2. From **Your applications**, choose **AWS Account**.
3. From the list of accounts, choose the account ID for your management account. This ID may also have a label, for example, **(Management)**.
4. From **AWSServiceCatalogEndUserAccess**, choose **Management console**. This opens the AWS Management Console for this user in this account.

5. Ensure that you've selected the correct AWS Region for provisioning accounts, which should be your AWS Control Tower Region.
6. Search for and choose **Service Catalog** to open the Service Catalog console.
7. In the navigation pane, choose **Products**.
8. Select **AWS Control Tower Account Factory**, then choose the **Launch product** button. This selection starts the wizard to provision a new account.
9. Fill in the information, and keep the following in mind:
   - The **SSOUserEmail** can be a new email address, or the email address associated with an existing IAM Identity Center user. Whichever you choose, this user will have administrative access to the account you're provisioning.
   - The **AccountEmail** must be an email address that isn't already associated with an AWS account. If you used a new email address in **SSOUserEmail**, you can use that email address here.
10. Don't define **TagOptions** and don't enable **Notifications**, otherwise the account can fail to be provisioned. When you're finished, choose **Launch product**.
11. Review your account settings, and then choose **Launch**. Don't create a resource plan, otherwise the account will fail to be provisioned.
12. Your account is now being provisioned. It can take a few minutes to complete. You can refresh the page to update the displayed status information.

   Note

   Up to five accounts can be provisioned at a time.

Considerations for managing accounts in Account Factory

You can update, unenroll, and close accounts that you create and provision through Account Factory. You can recycle accounts by updating the user parameters in the accounts that you want to repurpose. You can also change an account's organizational unit (OU).

Note

When updating a provisioned product that's associated with an account that Account Factory vends, if you specify a new user email address for AWS IAM Identity Center, AWS

Control Tower creates a new user in IAM Identity Center. The previously created account isn't removed. For information about removing the previous IAM Identity Center user email address from IAM Identity Center, see Disabling a User.

Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog

The easiest way to update an enrolled account is through the AWS Control Tower console. Individual account updates are useful for resolving drift, such as Moved member account. Account updates also are required as part of a full landing zone update.

If you move an account from one organizational unit (OU) to another, remember that the controls applied by the new OU may be different than the controls in the former OU. Be sure that the controls in the new OU meet your policy requirements for the account.

Control behavior when accounts are moved between OUs

When you move an account between OUs, the controls for the destination OU are applied to the account. However, the controls that applied to the account from the former OU are not removed. The exact behavior of the controls is specific to the implementation of the controls that are active on the former OU and the destination OU.

- *For controls implemented with AWS Config rules:* The controls from the previous OU are not removed. These controls must be removed manually.
- *For controls implemented with SCPs:* The SCP-based controls from the previous OU are removed. The SCP-based controls for the destination OU go into effect on this account.
- *For controls implemented with AWS CloudFormation hooks:* This behavior depends on the status of controls in the new OU.
  - *If the destination OU has no hook-based controls active:* The old controls remain active for the moved account, unless you remove them manually.
  - *If the destination OU has hook controls active:* The old controls are removed and the controls in the destination OU are applied to the account.

Update the account in the console

To update an account in the AWS Control Tower console

1. When signed in to AWS Control Tower, navigate to the **Organization** page.
2. In the list of OUs and accounts, select the name of the account you wish to update. Accounts that are available for updating show a status of **Update available**.
3. Next you'll see the **Account details** page for your selected account.
4. In the upper right, choose **Update account**.

Update the provisioned product

The following procedure guides you through how to update your account in Account Factory or move it to a new OU, by updating the account's provisioned product in Service Catalog.

To update an Account Factory account or change its OU through Service Catalog

1. Sign in to the AWS Management Console, and open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/.

   > **Note**
   >
   > You must sign in as a user with permissions to provision new products in Service Catalog (for example, an IAM Identity Center user in `AWSAccountFactory` or `AWSServiceCatalogAdmins` groups).

2. In the navigation pane, choose **Provisioning**, and then choose **Provisioned products**.
3. For each of the member accounts listed, perform the following steps to update all member accounts:

   a. Select a member account. You're directed to the *Provisioned product details* page for that account.

   b. On the *Provisioned product details* page, choose the **Events** tab.

   c. Make a note of the following parameters:

      - **SSOUserEmail** (Available in provisioned product details)
      - **AccountEmail** (Available in provisioned product details)

- **SSOUserFirstName** (Available in IAM Identity Center)
- **SSOUSerLastName** (Available in IAM Identity Center)
- **AccountName** (Available in IAM Identity Center)

d. From **Actions**, choose **Update**.

e. Choose the button next to the **Version** of the product you want to update, and choose **Next**.

f. Provide the parameter values that were mentioned previously.

- If you want to keep the existing OU, for **ManagedOrganizationalUnit**, choose the OU that the account was already in.
- If you want to migrate the account to a new OU, for **ManagedOrganizationalUnit**, choose the new OU for the account.

A central cloud administrator can find this information in the AWS Control Tower console, on the **Organization** page.

g. Choose **Next**.

h. Review your changes, and then choose **Update**. This process can take a few minutes per account.

Change email address of an enrolled account

To change the email address of an enrolled member account in AWS Control Tower, follow the procedure in this section.

> **Note**
>
> The following procedure doesn't allow you to change the email address of a **management account**, **log archive account**, or **audit account**. For more information about that, see How do I change the email address associated with my AWS account? or contact AWS Support.

To change the email address of an account that AWS Control Tower creates

1. Recover the root user password for the account. You can follow the steps in the article How do I recover a lost or forgotten AWS password?

2. Sign in to the account with the root user password.
3. Change the email address as you would for any other AWS account, and wait for the change to reflect in AWS Organizations. You might experience a delay while the email address change finishes updating.
4. Update the provisioned product in Service Catalog using the email address that previously belonged to the account. The process for updating the provisioned product includes associating the new email address with the provisioned product. This way the email address change takes effect in AWS Control Tower. Use the new email address for updates to subsequently provisioned products.

To change the password or email address of a member account that you created with AWS Organizations, see Accessing a member account as the root user in the *AWS Organizations User Guide*.

Alternatively, you can update the email address for an Account Factory or other member account from the AWS Organizations console without logging in as the root user. For more information, see Updating the root user email address for a member account with AWS Organizations in the *AWS Organizations User Guide*.

Change the name of an enrolled account

Follow the procedure in this section to change the name of an enrolled AWS Control Tower account.

> **Note**
>
> To change the name of an AWS *administrator* account, you must have admin permissions and be logged in as the account's root user.

To change the name of an account created by AWS Control Tower

1. Recover the root password for the account. You can follow the steps outlined in this article, How do I recover a lost or forgotten AWS password?
2. Sign in to the account with the root password.
3. In the AWS Billing console, navigate to the **Account settings** page.
4. Change the name in **Account settings**, as you would for any other AWS account.

5. AWS Control Tower automatically updates itself to reflect the name change. This update will not be reflected in the provisioned product in AWS Service Catalog.

Configure Account Factory with Amazon Virtual Private Cloud settings

Account Factory allows you to create pre-approved baselines and configuration options for accounts in your organization. You can configure and provision new accounts through AWS Service Catalog.

On the Account Factory page, you can see a list of organizational units (OUs) and their **allow list** status. By default, all OUs are on the allow list, which means that accounts can be provisioned under them. You can disable certain OUs for account provisioning through AWS Service Catalog.

You can view the Amazon VPC configuration options available to your end users when they provision new accounts.

To configure Amazon VPC settings in Account Factory

1. As a central cloud administrator, sign into the AWS Control Tower console with administrator permissions in the management account.
2. From the left side of the dashboard, select **Account Factory** to navigate to the Account Factory network configuration page. There you can see the default network settings displayed. To edit, select **Edit** and view the editable version of your Account Factory network configuration settings.
3. You can modify each field of the default settings as needed. Choose the VPC configuration options you'd like to establish for all new Account Factory accounts that your end users may create, and enter your settings into the fields.

- Choose **disabled** or **enabled** to create a public subnet in Amazon VPC. By default, the internet-accessible subnet is disallowed.

> **Note**
> If you set the account factory VPC configuration so that public subnets are **enabled** when provisioning a new account, account factory configures Amazon VPC to create a NAT Gateway. You will be billed for your usage by Amazon VPC. See VPC Pricing for more information.

- Choose the maximum number of private subnets in Amazon VPC from the list. By default, 1 is selected. The maximum number of private subnets allowed is 2 per availability zone.
- Enter the range of IP addresses for creating your account VPCs. The value must be in the form of a classless inter-domain routing (CIDR) block (for example, the default is `172.31.0.0/16`). This CIDR block provides the overall range of subnet IP addresses for the VPC that Account Factory creates for your account. Within your VPC, subnets are assigned automatically from the range you specify, and they are equal in size. By default, subnets within your VPC do not overlap. However, subnet IP address ranges in the VPCs of all your provisioned accounts could overlap.
- Choose a region or all the regions for creating a VPC when an account is provisioned. By default all available regions are selected.
- From the list, choose the number of Availability Zones to configure subnets for in each VPC. The default and recommended number is 3.
- Choose **Save**.

You can set up these configuration options to create new accounts that don't include a VPC. See the walkthrough.

Unenroll an account

If you created an account in Account Factory or enrolled an AWS account, and you no longer want the account to be managed by AWS Control Tower in a landing zone, you can *unenroll* the account from the AWS Control Tower console.

When you unenroll an AWS Control Tower account, all resources provisioned by AWS Control Tower are removed, including any blueprints. The account is moved out of any AWS Control Tower OU and into the **Root** area. The account is no longer part of a registered OU, and it is no longer subject to AWS Control Tower SCPs. You can close the account through AWS Organizations.

To unenroll an enrolled account from the AWS Control Tower console

1. Open the AWS Control Tower console in your web browser at https://console.aws.amazon.com/controltower
2. In the left navigation pane, choose **Organization**.
3. In the **Organization** page, expand the OU that contains the account, by selecting the **+** button near the OU.
4. Select the account and then choose **Unmanage**.

Unenrolling an account also can be done in the Service Catalog console by an IAM Identity Center user in the `AWSAccountFactory` group, by terminating the Provisioned Product. For more information on IAM Identity Center users or groups, see [Manage users and access through AWS IAM Identity Center](). The following procedure describes how to unenroll a member account in Service Catalog.

To unenroll an enrolled account through Service Catalog

1. Open the Service Catalog console in your web browser at https://console.aws.amazon.com/servicecatalog.
2. In the left navigation pane, choose **Provisioned products list**.
3. From the list of provisioned accounts, choose the name of the account that you want AWS Control Tower no longer to manage.
4. On the **Provisioned product details** page, from the **Actions** menu, choose **Terminate**.
5. From the dialog box that appears, choose **Terminate**.

> **Important**
>
> The word *terminate* is specific to Service Catalog. When you terminate an account in Service Catalog Account Factory, the account is not closed. This action removes the account from its OU and your landing zone.

6. When the account has been unenrolled, its status changes to **Not Enrolled**.
7. If you no longer need the account, close it. For more information about closing AWS accounts, see Closing an account in the *AWS Billing User Guide*

When you unenroll a customized account, AWS Control Tower removes the resources that the blueprint has deployed, as well as any other resources that AWS Control Tower created within the account. After you unenroll the account, you can close the account through AWS Organizations.

> **Note**
>
> An unenrolled account is not closed or deleted. When the account has been unenrolled, the IAM Identity Center user that you selected when you created the account in Account Factory still has administrative access to the account. If you do not want this user to have administrative access, you must change this setting in IAM Identity Center by updating the account in Account Factory and changing the IAM Identity Center user email address for the

account. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.

Video walkthrough

This video (3:25) describes how to remove an account from AWS Control Tower, gain root access to the account, and finally close the AWS account. You also can close an account with an AWS Organizations API. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Closing an Account in AWS Control Tower.

You can view a list of AWS YouTube videos that explain common tasks in AWS Control Tower.

Close an account created in Account Factory

Accounts created in Account Factory are AWS accounts. For information about closing AWS accounts, see Closing an account in the *AWS Account Management Reference Guide*.

> **Note**
>
> Closing an AWS account is not the same as unenrolling an account from AWS Control Tower —these are separate actions. You must unenroll the account before you close it.

Close an AWS Control Tower member account through AWS Organizations

You can close your AWS Control Tower member accounts from your organization's management account without a requirement to sign in to each member account individually with root credentials, by means of AWS Organizations. You cannot close your management account in this way, however.

When you call the AWS Organizations CloseAccount API, or close an account in the AWS Organizations console, the member account is isolated for 90 days, as any AWS account would be. The account shows a **Suspended** status in AWS Control Tower and AWS Organizations. If you attempt to work with the account during that 90 days, AWS Control Tower gives an error message.

Before the 90 days expire, you can restore the member account, as you can do with any AWS account. After that 90-day time, the account's records are removed.

We recommend, as a best practice, to unenroll a member account before you close that account. If you close a member account without first unmanaging it, AWS Control Tower shows the account's status as **Suspended**, but also as **Enrolled**. As a result, if you attempt to **Re-register** the account's OU during that 90-day time, AWS Control Tower produces an error message. The suspended account essentially blocks the re-registering actions with a pre-check failure. If you remove the account from the OU, you can **Re-register** the OU, but AWS may produce an error regarding a missing method of payment for the account. To work around this constraint, create another OU, and move the account to that OU before you try to re-register. We recommend naming this OU the **Suspended** OU.

> **Note**
>
> If you do not unenroll the account before you close it, you must delete the account's provisioned product in AWS Service Catalog after those 90 days are finished.

For more information, see the AWS Organizations documentation about the [CloseAccount API](#).

Resource Considerations for Account Factory

When an account is provisioned with Account Factory, the following AWS resources are created within the account.

AWS service	Resource type	Resource name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-* StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-* StackSet-AWSControlTowerBP-BASELINE-CONFIG-* StackSet-AWSControlTowerBP-BASELINE-ROLES-*

AWS service	Resource type	Resource name
		StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES-*
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs /aws/lambda/aws-controltower-NotificationForwarder
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole AWSControlTowerExecution
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy

AWS service	Resource type	Resource name
Amazon Simple Notification Service	Topics	aws-controltower-SecurityNotifications
AWS Lambda	Applications	StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*
AWS Lambda	Functions	aws-controltower-NotificationForwarder
Amazon EventBridge	Rule	AWSControlTowerManagedRule
Amazon EventBridge	Rule	aws-controltower-ConfigComplianceChangeEventRule

Customize accounts with Account Factory Customization (AFC)

AWS Control Tower allows you to customize new and existing AWS accounts when you provision their resources from the AWS Control Tower console. After you set up Account Factory customization, AWS Control Tower automates this process for future provisioning, so you don't have to maintain any pipelines. Customized accounts are available for use immediately after the resources are provisioned.

Provision new accounts with blueprints

Your customized accounts are provisioned in the AWS Control Tower Account Factory, through AWS CloudFormation templates, or with Terraform. You'll define a template that serves as customized account *blueprint*. Your blueprint describes the specific resources and configurations you require when an account is provisioned. Pre-defined blueprints, built and managed by AWS partners, also are available. For more information about partner-managed blueprints, see the AWS Service Catalog Getting Started Library.

Apply blueprints to existing accounts

You can apply customized blueprints to existing accounts, also, by following the **Update account** steps in the AWS Control Tower console. For details, see Update the account in the console.

Definition: Your hub account

Your account blueprints are stored in an AWS account, which for our purposes is referred to as a *hub account*. Blueprints are stored in the form of an Service Catalog product. We call this product a blueprint, to distinguish it from any other Service Catalog products. To learn more about how to create Service Catalog products, see Creating products in the *AWS Service Catalog Administrator Guide*.

> **Note**
>
> AWS Control Tower contains *proactive controls*, which monitor AWS CloudFormation resources in AWS Control Tower. Optionally, you can activate these controls in your landing zone. When you apply proactive controls, they check to make sure that the resources you're about to deploy to your accounts are compliant with your organization's policies and procedures. For more information about proactive controls, see Proactive controls.

For more information about working with AFC, see Automate account customization using Account Factory Customization in AWS Control Tower.

> **Prerequisites**
>
> Before you begin to create customized accounts with AWS Control Tower Account Factory, you must have an AWS Control Tower landing zone environment deployed, and you must have an organizational unit (OU) registered with AWS Control Tower, where your newly created accounts will be placed.

Preparation for customization

- *Designate a hub account:* You may create a new account to serve as the hub account, or you may use an existing AWS account. We strongly recommend that you do not use the AWS Control Tower management account as your blueprint hub account.
- *Add the necessary role:* If you plan to enroll AWS accounts into AWS Control Tower and customize them, you must first add the `AWSControlTowerExecution` role to those accounts, as you would for any other account you are enrolling into AWS Control Tower.

- *Configure partner blueprints (optional):* If you plan to use partner blueprints that have marketplace subscription requirements, you must configure these from your AWS Control Tower management account before you deploy the partner blueprints as account factory customization blueprints.

Topics

Set up for customization

The next sections give steps to set up Account Factory for the customization process. We recommend that you set up delegated admin for the hub account, before you begin these steps.

Summary

- **Step 1. Create the required role.** Create an IAM role that grants permission for AWS Control Tower to have access to the (hub) account, where the Service Catalog products, also called blueprints, are stored.
- **Step 2. Create the AWS Service Catalog product.** Create the AWS Service Catalog product (also called a “blueprint product”) that you'll need for baselining the custom account.
- **Step 3. Review your custom blueprint.** Inspect the AWS Service Catalog product (blueprint) that you created.

- **Step 4. Call your blueprint to create a customized account.** Enter the blueprint product information and the role information into the proper fields in Account Factory, in the AWS Control Tower console, while creating the account.

Step 1. Create the required role

Before you begin to customize accounts, you must set up a role that contains a trust relationship between AWS Control Tower and your hub account. When assumed, the role grants AWS Control Tower access to administer the resources in the hub account. The role must be named **AWSControlTowerBlueprintAccess**.

AWS Control Tower assumes this role to create a Portfolio resource on your behalf in AWS Service Catalog, then to add your blueprint as a Service Catalog Product to this Portfolio, and then to share this Portfolio, and your blueprint, with your member account during account provisioning.

You'll create the `AWSControlTowerBlueprintAccess` role, as explained in the following sections. You can set up the role in an enrolled or an unenrolled account.

> **Navigate to the IAM console to set up the required role.**

To set up the AWSControlTowerBlueprintAccess role in an enrolled AWS Control Tower account

1. Federate or sign in as the principal in the AWS Control Tower management account.
2. From the federated principal in the management account, assume or switch roles to the `AWSControlTowerExecution` role in the enrolled AWS Control Tower account that you select to serve as the blueprint hub account.
3. From the `AWSControlTowerExecution` role in the enrolled AWS Control Tower account, create the `AWSControlTowerBlueprintAccess` role with proper permissions and trust relationships.

> **Important**
>
> To comply with AWS best practices guidance, it's important that you sign out of the `AWSControlTowerExecution` role immediately after you create the `AWSControlTowerBlueprintAccess` role.

To prevent unintended changes to resources, the `AWSControlTowerExecution` role is intended for use by AWS Control Tower only.

If your blueprint hub account isn't enrolled in AWS Control Tower, the `AWSControlTowerExecution` role won't exist in the account, and there's no need to assume it before you continue with setting up the `AWSControlTowerBlueprintAccess` role.

To set up the AWSControlTowerBlueprintAccess role in an unenrolled member account

1. Federate or sign in as a principal in the account that you wish to designate as the hub account, by means of your preferred method.
2. When signed in as the principal in the account, create the `AWSControlTowerBlueprintAccess` role with proper permissions and trust relationships.

The **AWSControlTowerBlueprintAccess** role must be set up to grant trust to two principals:

- The principal (user) that runs AWS Control Tower in the AWS Control Tower management account.
- The role named `AWSControlTowerAdmin` in the AWS Control Tower management account.

Here's an example trust policy, similar to one you will need to include for your role. This policy demonstrates the best practice of granting least-privilege access. When you make your own policy, replace the term *`YourManagementAccountId`* with the actual acccount ID of your AWS Control Tower management account, and replace the term *`YourControlTowerUserRole`* with the identifier of the IAM role for your management account.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/service-role/
AWSControlTowerAdmin",
```

```
                    "arn:aws:iam::111122223333:role/YourControlTowerUserRole"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Required permissions policy

AWS Control Tower requires that the managed policy named `AWSServiceCatalogAdminFullAccess` must be attached to the `AWSControlTowerBlueprintAccess` role. This policy provides permissions that AWS Service Catalog looks for when it allows AWS Control Tower to administer your portfolio and AWS Service Catalog Product resources. You can attach this policy when you're creating the role in the IAM console.

> **Additional permissions may be required**
>
> - If you store your blueprints in Amazon S3, AWS Control Tower also requires the `AmazonS3ReadOnlyAccess` permission policy for the `AWSControlTowerBlueprintAccess` role.
> - The AWS Service Catalog Terraform type of product requires you to add some additional permissions to the AFC custom IAM policy, if you don't utilize the default **Admin** policy. It requires these in addition to the permissions required to create the resources that you define in your terraform template.

Step 2. Create the AWS Service Catalog product

To create an AWS Service Catalog product, follow the steps at [Creating products](#) in the *AWS Service Catalog Administrator Guide*. You'll add your account blueprint as a template when you create the AWS Service Catalog product.

> **Important**
>
> As a result of HashiCorp's updated Terraform licensing, AWS Service Catalog changed support for *Terraform Open Source* products and provisioned products to a new product

type, called *External*. To learn more about how this change effects AFC, including how to update your existing account blueprints to the External product type, review Transition to External product type.

Summary of steps to create a blueprint

- Create or download an AWS CloudFormation template or Terraform tar.gz configuration file that will become your account blueprint. Some template examples are given later in this section.
- Sign in to the AWS account where you store your Account Factory blueprints (sometimes called the hub account).
- Navigate to the AWS Service Catalog console. Choose **Product list**, and then choose **Upload new product**.
- In the **Product details** pane, enter details for your blueprint product, such as a name and description.
- Select **Use a template file** and then select **Choose file**. Select or paste the template or configuration file you've developed or downloaded for use as your blueprint.
- Choose **Create product** at the bottom of the console page.

You can download an AWS CloudFormation template from the AWS Service Catalog reference architecture repository. One example from that repository helps set up a backup plan for your resources.

Here's an example template, for a fictitious company called **Best Pets**. It helps set up a connection to their pet database.

```
Resources:
  ConnectionStringGeneratorLambdaRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
            Action:
```

```
            - "sts:AssumeRole"
  ConnectionStringGeneratorLambda:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Join ['-', ['ConnectionStringGenerator', !Select [4, !Split
 ['-', !Select [2, !Split ['/', !Ref AWS::StackId]]]]]]
      Description: Retrieves the connection string for this account to access the Pet
 Database
      Role: !GetAtt ConnectionStringGeneratorLambdaRole.Arn
      Runtime: nodejs22.x
      Handler: index.handler
      Timeout: 5
      Code:
        ZipFile: >
           export const handler = async (event, context) => {
             const awsAccountId = context.invokedFunctionArn.split(“:”)[4]
             const connectionString= “fake connection for account ” + awsAccountId;
             const response = {
               statusCode: 200,
               body: connectionString
             };
           return response;
          };

  ConnectionString:
    Type: Custom::ConnectionStringGenerator
    Properties:
      ServiceToken: !GetAtt ConnectionStringGeneratorLambda.Arn

  PetDatabaseConnectionString:
    DependsOn: ConnectionString
    # For example purposes we're using SSM parameter store.
    # In your template, use secure alternatives to store
    # sensitive values such as connection strings.
    Type: AWS::SSM::Parameter
    Properties:
      Name: pet-database-connection-string
      Description: Connection information for the BestPets pet database
      Type: String
      Value: !GetAtt ConnectionString.Value
```

Step 3. Review your custom blueprint

You can view your blueprint in the AWS Service Catalog console. For more information, see Managing products in the *Service Catalog Adminstrator Guide*.

Step 4. Call your blueprint to create a customized account

When you follow the **Create account** workflow in the AWS Control Tower console, you'll see an optional section where you can enter information about the blueprint you'd like to use for customizing accounts.

> **Prerequisites**
>
> You must set up your customization hub account and add at least one blueprint (Service Catalog product) before you can enter that information into the AWS Control Tower console and begin to provision customized accounts.

Create or update a customized account in the AWS Control Tower console.

1. Enter the account ID for the account that contains your blueprints.
2. From that account, select an existing Service Catalog product (existing blueprint).
3. Select the proper version of the blueprint (Service Catalog product), if you have more than one version.
4. (Optional) You can add or change a blueprint provisioning policy at this point in the process. The blueprint provisioning policy is written in JSON and attached to an IAM role, so it can provision the resources that are specified in the blueprint template. AWS Control Tower creates this role in the member account so that Service Catalog can deploy resources using AWS CloudFormation stack sets. The role is named `AWSControlTower-BlueprintExecution-bp-xxxx`. The `AdministratorAccess` policy is applied here by default.
5. Choose the AWS Region or Regions in which you wish to deploy accounts based on this blueprint.
6. If your blueprint contains parameters, you can enter the values for the parameters into additional fields in the AWS Control Tower workflow. The additional values may include: a GitHub repository name, a GitHub branch, an Amazon ECS cluster name, and a GitHub identity for the repository owner.

7. You can customize accounts at a later time by following the **Account update** process, if your hub account or blueprints are not yet ready.

For more details, see Create a customized account from a blueprint.

Create a customized account from a blueprint

After you have created custom blueprints, you can start creating custom accounts in AWS Control Tower account factory.

Follow these steps to deploy a custom blueprint when you're creating a new AWS account:

1. Go to AWS Control Tower in the AWS Management Console.
2. Select **Account factory** and **Create account**.
3. Enter account details such as account name and email address.
4. Configure IAM Identity Center details with email address and user name.
5. Select a registered OU where your account will be added.
6. Expand the **Account factory customization** section.
7. Enter the account ID of the blueprint hub account that contains your Service Catalog products and choose **Validate**. For more information about a blueprint hub account, see Customize accounts with Account Factory Customization (AFC).
8. Select the dropdown menu that contains all blueprints from your Service Catalog Product List (all custom and partner blueprints). Choose a blueprint and corresponding version to deploy.
9. If your blueprint contains parameters, these fields are displayed for you to populate. Default values are pre-populated.
10. Finally, select where you'll deploy your blueprint, either **Home Region** or **All governed Regions**. Global resources such as Route 53 or IAM, may need to be deployed to a single Region only. Regional resources, such as Amazon EC2 instances or Amazon S3 buckets, could be deployed to all governed Regions
11. After all fields are completed, select **Create account**.

> **Note**
>
> Blueprints created with Terraform can deploy to one Region only, not multiple Regions.

You can view the progress of your account provisioning on the **Organization** page. When your account provisioning is complete, the resources specified by your blueprint are already deployed within it. To view the details of the account and blueprint, go to the **Account details** page.

Customize accounts with AFC as you enroll them

To enroll and customize accounts in the AWS Control Tower console.

1. Navigate to the AWS Control Tower console and select **Organization** from the left navigation.
2. You will see a list of your available accounts. Identify the account you would like to enroll with a custom blueprint. The **State** column for that account should reflect the account in a **Not enrolled** status.
3. Select the radio button to the left of the account and choose the **Actions** dropdown menu, in the top right of the screen. Here you will select the **Enroll** option.
4. Complete the **Access configuration** section with the account's IAM Identity Center information.
5. Select the registered OU where your account will become a member.
6. Complete the **Account factory customization** section using the same steps as 7-12 of the **Create account** procedure. For more information, see Provision Account Factory accounts with AWS Service Catalog.

You can view the status of your account progress on the **Organization** page. When your account enrollment is complete, the resources specified by the blueprint are already deployed within it.

Add a blueprint to an AWS Control Tower account

To add a blueprint to an existing AWS Control Tower member account, follow the **Update account** workflow in the AWS Control Tower console, and choose a new blueprint to add to the account. For more information, see Update and move Account Factory accounts with AWS Control Tower or with AWS Service Catalog.

> **Note**
>
> If you add a new blueprint to an account, the existing blueprint is overwritten.

Note

One blueprint may be deployed per AWS Control Tower account.

Update a blueprint

The following procedures describe how to update custom blueprints and how to deploy them.

To update your custom blueprints

1. Update your AWS CloudFormation template or Terraform tar.gz file (blueprint) with your new configurations.
2. Save the updated blueprint as a new version in AWS Service Catalog.

To deploy your updated blueprint

1. Navigate to the **Organization** page in the AWS Control Tower console.
2. Filter the **Organization** page by blueprint name and version.
3. Follow the **Update account** process, and deploy the latest blueprint version in your account.

If a blueprint update is unsuccessful

AWS Control Tower allows blueprint updates when the provisioned product is in the `AVAILABLE` state. If your provisioned product is in a `TAINTED` state, the update will fail. We recommend the following workaround:

1. In the AWS Service Catalog console, manually update the `TAINTED` provisioned product to change the state to `AVAILABLE`. For more information, see Updating provisioned products.
2. Then, follow the update account process from AWS Control Tower to fix the blueprint deployment error.

We recommend this manual step because: When you remove a blueprint, it can cause resources in the member account to be removed. Removing resources may affect your existing workloads. For this reason, we recommend this method rather than the alternative way of updating a blueprint—which is by removing and replacing the original blueprint—especially if you are running production workloads.

Remove a blueprint from an account

To remove a blueprint from an account, follow the **Update account** workflow to remove the blueprint and return the account to the AWS Control Tower default configurations.

As you enter the **Update account** workflow in the console, you will see that all of the account details are populated, and the customization details are not populated. If you leave these AFC details blank, AWS Control Tower removes the blueprint from the account. You will see a warning message before the action begins.

> **Note**
>
> AWS Control Tower adds a blueprint to an account only if you select a blueprint during the **Create account** or **Update account** process.

Partner blueprints

AWS Control Tower Account Factory Customization (AFC) provides access to pre-defined customization blueprints that are built and managed by AWS Partners. These partner blueprints help you customize your accounts for specific use cases. Each partner's blueprints help you build customized accounts, which are pre-configured to work with the product offerings from that particular partner.

To view a complete list of AWS Control Tower partner blueprints, navigate to the Service Catalog **Getting Started Library** in your console. Search for the source type **AWS Control Tower Blueprints**.

Considerations for Account Factory Customizations (AFC)

- AFC supports customization using a single AWS Service Catalog blueprint product only.
- The AWS Service Catalog blueprint products must be created in the hub account, and in the same Region as the AWS Control Tower landing zone home Region.
- The `AWSControlTowerBlueprintAccess` IAM role must be created with the proper name, permissions, and trust policy.
- AWS Control Tower supports two deployment options for blueprints: deploy to the home Region only, or deploy to all Regions governed by AWS Control Tower. Selection of Regions is not available.

- When you update a blueprint in a member account, the blueprint hub account ID and the AWS Service Catalog blueprint product cannot be changed.
- AWS Control Tower doesn't support removing an existing blueprint and adding a new blueprint in a single blueprint update operation. You can remove a blueprint and then add a new blueprint in separate operations.
- AWS Control Tower changes behavior, based on whether you are creating or enrolling customized accounts, or non-customized accounts. If you are not creating or enrolling customized accounts with blueprints, AWS Control Tower creates an Account Factory provisioned product (through Service Catalog) in the AWS Control Tower management account. If you are specifying customization when creating or enrolling accounts with blueprints, AWS Control Tower does not create an Account Factory provisioned product in the AWS Control Tower management account.

In case of a blueprint error

Error while applying a blueprint

If an error occurs during the process of applying a blueprint to an account—either a new account or an existing account that you are enrolling into AWS Control Tower—the recovery procedure is the same. The account will exist, but it is not customized, and it is not enrolled into AWS Control Tower. To continue, follow the steps to enroll the account into AWS Control Tower, and add the blueprint at time of enrollment.

Error while creating the `AWSControlTowerBlueprintAccess` role, and workarounds

When you create the `AWSControlTowerBlueprintAccess` role from an AWS Control Tower account, you must be signed in as the principal using the `AWSControlTowerExecution` role. If you are signed in as any other, the `CreateRole` operation is prevented by an SCP, as shown in the artifact that follows:

```
{
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:role/AWSControlTowerExecution",
                        "arn:aws:iam::*:role/stacksets-exec-*"
                    ]
                }
            },
```

```
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:DeleteRolePermissionsBoundary",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:PutRolePermissionsBoundary",
                "iam:PutRolePolicy",
                "iam:UpdateAssumeRolePolicy",
                "iam:UpdateRole",
                "iam:UpdateRoleDescription"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-controltower-*",
                "arn:aws:iam::*:role/*AWSControlTower*",
                "arn:aws:iam::*:role/stacksets-exec-*"
            ],
            "Effect": "Deny",
            "Sid": "GRIAMROLEPOLICY"
        }
```

The following workarounds are available:

- (Most recommended) Assume the `AWSControlTowerExecution` role and create the `AWSControlTowerBlueprintAccess` role. If you choose this workaround, be sure to sign out from the `AWSControlTowerExecution` role immediately afterward, to prevent unintended changes to resources.
- Sign into an account that is not enrolled in AWS Control Tower, and therefore not subject to this SCP.
- Temporarily edit this SCP to permit the operation.
- (Strongly not recommended) Use your AWS Control Tower management account as your hub account, so it is not subject to the SCP.

Customizing your policy document for AFC blueprints based on CloudFormation

When you enable a blueprint through account factory, AWS Control Tower directs AWS CloudFormation to create a StackSet on your behalf. AWS CloudFormation requires access to your managed account to create AWS CloudFormation stacks in the StackSet. Although AWS CloudFormation already has administrator privileges in the managed account through the `AWSControlTowerExecution` role, this role is not assumable by AWS CloudFormation.

As part of enabling a blueprint, AWS Control Tower creates a role in the member account, which AWS CloudFormation may assume to complete the StackSet management tasks. The simplest way to enable your customized blueprint through account factory is to use an *allow-all* policy, because those policies are compatible with any blueprint template.

However, best practices suggest that you must restrict the permissions for AWS CloudFormation in the target account. You can provide a customized policy, which AWS Control Tower applies to the role it creates for AWS CloudFormation to use. For example, if your blueprint creates an SSM Parameter called *something-important*, you could provide the following policy:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudFormationActionsOnStacks",
            "Effect": "Allow",
            "Action": "cloudformation:*",
            "Resource": "arn:aws:cloudformation:*:*:stack/*"
        },
        {
            "Sid": "AllowSsmParameterActions",
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                 "ssm:DeleteParameter",
                 "ssm:GetParameter",
                 "ssm:GetParameters"
            ],
            "Resource": "arn:*:ssm:*:*:parameter/something-important"
        }
    ]
}
```

The `AllowCloudFormationActionsOnStacks` statement is required for all AFC custom policies; AWS CloudFormation uses this role to create stack instances, therefore it requires permission to perform AWS CloudFormation actions on stacks. The `AllowSsmParameterActions` section is specific to the template being enabled.

Resolve permission issues

When you enable a blueprint with a restricted policy, you may find that there are insufficient permissions to enable the blueprint. To resolve these issues, revise your policy document and update the member account's blueprint preferences to use the corrected policy. To check that the policy is sufficient to enable the blueprint, ensure that the AWS CloudFormation permissions are granted, and that you can create a stack directly using that role.

Additional permissions required for creating a Terraform-based Service Catalog product

When you're creating an AWS Service Catalog External product with a Terraform configuration file for AFC, AWS Service Catalog requires certain permissions to be added to your AFC custom IAM policy, in addition to permissions required to create the resources defined in your template. If you choose the default full **Admin** policy, you do not need to add these extra permissions.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:ListGroupResources",
                "resource-groups:DeleteGroup",
                "resource-groups:Tag"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "tag:GetResources",
                "tag:GetTagKeys",
```

```
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "s3:GetObject",
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        }
    ]
}
```

For more information about creating Terraform products using the External product type in AWS Service Catalog, see Step 5: Create launch roles in the Service Catalog Administrator Guide.

Provision accounts with AWS Control Tower Account Factory for Terraform (AFT)

AWS Control Tower Account Factory for Terraform (AFT) adopts a GitOps model that automates the process of account provisioning and updating in AWS Control Tower.

> **Note**
>
> AFT doesn't impact workflow performance in AWS Control Tower. If you provision an account through AFT or Account Factory, the same backend workflow occurs.

With AFT, you create an account request Terraform file, which contains the input that invokes the AFT workflow. After account provisioning and updating finishes, the AFT workflow continues by running the AFT account provisioning framework and account customizations steps.

Prerequisites

When you're getting started with AFT, you will create the following:

- In AWS Control Tower create the OU, and then the AFT management account, for your AFT environment. Make note of the account ID, so you can enter it in the `main.tf` file later, when you deploy AFT with the Terraform module. You can view this account ID on the AWS Control Tower **Control details** page. For more information, see the Terraform documentation.
- One or more `git` repositories for your fully deployed AFT environment. For more information, see Post-deployment steps for AFT.
- A fully deployed AFT environment. For more information, see Overview of AWS Control Tower Account Factory for Terraform (AFT) and Deploy AWS Control Tower Account Factory for Terraform (AFT). Also see the Terraform documentation.

> **Tip**
>
> You can create the AFT management account from the AWS Control Tower console with **Create account**. For more information, see Methods of provisioning.
> Also, optionally, you can create an account template folder to help define your additional accounts, in the **aft-account-customizations** repository.

For information about AWS Regions where AFT has deployment limitations, see Limitations and quotas in AWS Control Tower and Control limitations.

The Terraform documentation contains a good overview of how to set up AWS Control Tower Account Factory for Terraform (AFT).

Overview of AWS Control Tower Account Factory for Terraform (AFT)

Account Factory for Terraform (AFT) sets up a Terraform pipeline to help you provision and customize accounts in AWS Control Tower. AFT provides you with the advantage of Terraform-based account provisioning while allowing you to govern your accounts with AWS Control Tower.

With AFT you create an *account request Terraform file* to get the input that triggers the AFT workflow for account provisioning. After the account provisioning stage is complete, AFT automatically runs a series of steps before the account customizations stage begins. For more information, see AFT account provisioning pipeline.

AFT supports Terraform Cloud, Terraform Enterprise, and Terraform Community Edition. With AFT you can initiate account creation using an input file and a simple `git push` command and customize new or existing accounts. Account creation includes all of the AWS Control Tower governance benefits and account customizations that help you meet your organization's standard security procedures and compliance guidelines.

AFT supports account customization request tracing. Every time you submit an account customization request, AFT generates a unique tracing token that passes through an AFT customizations AWS Step Functions state machine, which logs the token as part of its execution. You can then use Amazon CloudWatch Logs insights queries to search timestamp ranges and retrieve the request token. As a result, you can see payloads that accompany the token, so you can trace your account customization request throughout the entire AFT workflow. For information about CloudWatch Logs and Step Functions, see the following:

- What is Amazon CloudWatch Logs? in the *Amazon CloudWatch Logs User Guide*
- What is AWS Step Functions? in the *AWS Step Functions Developer Guide*

AFT combines the capabilities of other AWS services as Component services, to build a framework, with pipelines that deploy Terraform Infrastructure as Code (IaC). AFT enables you to:

- Submit account provisioning and update requests in a GitOps model
- Store account metadata and audit history
- Apply account-level tags
- Add customizations to all accounts, to a set of accounts, or to individual accounts
- Enable feature options

AFT creates a separate account, called the *AFT management account*, to deploy AFT capabilities. Before you can set up AFT, you must have an existing AWS Control Tower landing zone. The AFT management account is not the same as the AWS Control Tower management account.

AFT offers flexibility

- **Flexibility for your platform:** AFT supports any Terraform Distribution for initial deployment and ongoing operation: Community Edition, Cloud, and Enterprise.
- **Flexibility for your version control system:** AFT supports AWS CodeCommit, and alternative version control sources through AWS CodeConnections.

AFT offers feature options

You can enable several feature options, based on best practices:

- Creating an organization-level CloudTrail for logging data events
- Deleting the AWS default VPC for accounts
- Enrolling provisioned accounts into the AWS Enterprise Support plan

> **Note**
>
> The AFT pipeline is not intended for use in deploying resources, such as Amazon EC2 instances, that your accounts require to run your applications. It is intended solely for automated provisioning and customizing of AWS Control Tower accounts.

Video Walkthrough

This video (7:33) describes how to deploy accounts with AWS Control Tower Account Factory for Terraform. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Automated Account Provisioning in AWS Control Tower.

AFT Architecture

Order of operations

You run AFT operations in the AFT management account. For a full account provisioning workflow, the order of stages from left to right in the diagram are as follows:

1. Account requests are created and submitted to the pipeline. You can create and submit more than one account request at a time. Account Factory processes requests in a first-in-first-out order. For more information, see Submit multiple account requests.
2. Each account is provisioned. This stage runs in the AWS Control Tower management account.
3. Global customizations run in the pipelines that are created for each vended account.
4. If customizations are specified in the initial account provisioning requests, the customizations run only on targeted accounts. If you have an account that's already provisioned, you must initiate further customizations manually in the account's pipeline.

AWS Control Tower Account Factory for Terraform – account provisioning workflow

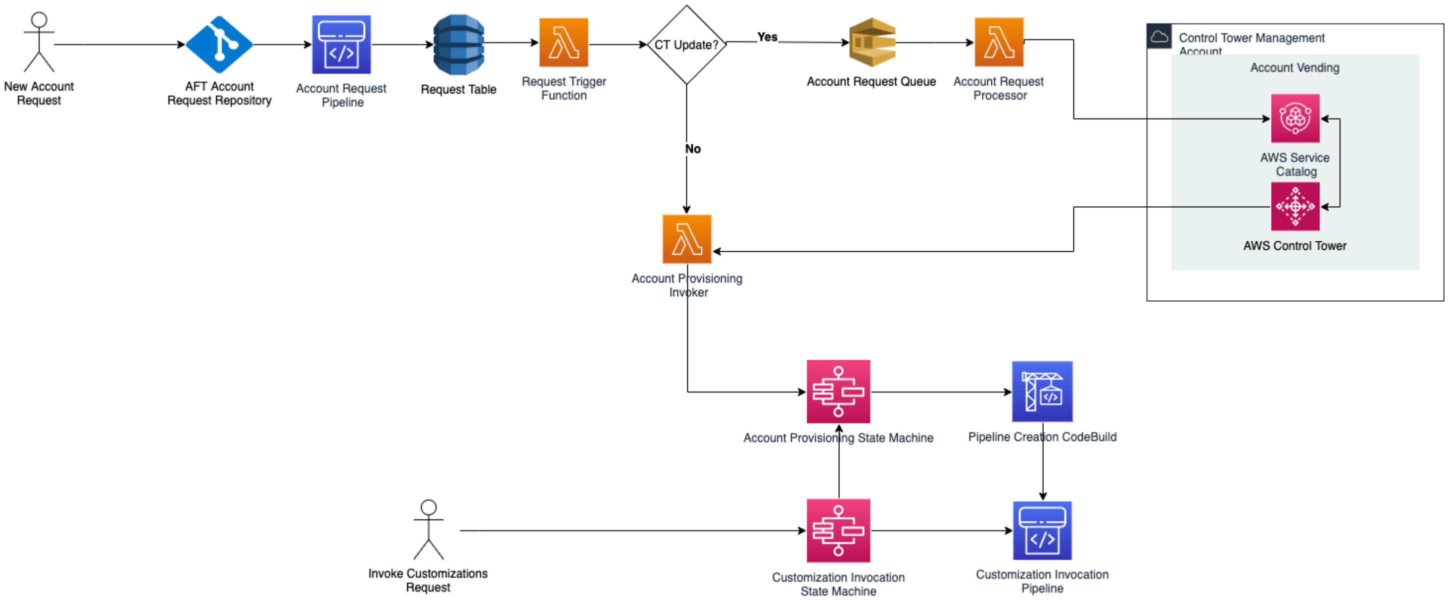

Figure 1: AWS Control Tower Account Factory for Terraform

Cost

No additional charge exists for AFT. You pay only for the resources deployed by AFT, the AWS services enabled by AFT, and the resources you deploy in your AFT environment.

The default AFT configuration includes the allocation of AWS PrivateLink endpoints, for enhanced data protection and security, and a NAT gateway that is required to support AWS CodeBuild. For details on the pricing of this infrastructure, see the AWS PrivateLink pricing and the Amazon VPC pricing for the NAT Gateway. Contact your AWS account representative for more specific information about managing these costs. You can change these default settings for AFT.

Deploy AWS Control Tower Account Factory for Terraform (AFT)

This section is for administrators of AWS Control Tower environments who wish to set up Account Factory for Terraform (AFT) in their existing environment. It describes how to set up an Account Factory for Terraform (AFT) environment with a new, dedicated AFT management account.

> **Note**
>
> A Terraform module deploys AFT. This module is available in the AFT repository on GitHub, and the entire AFT repository is considered the module.

We recommend that you refer to the AFT modules on GitHub instead of cloning the AFT repository. This way you can control and consume updates to the modules as they are available.

For details about the latest releases of the AWS Control Tower Account Factory for Terraform (AFT) functionality, see the Releases file for this GitHub repository.

Deployment prerequisites

Before you configure and launch your AFT environment, you must have the following resources available:

- A home Region for your AWS Control Tower landing zone. For more information, see How AWS Regions work with AWS Control Tower.
- An AWS Control Tower landing zone. For more information, see Plan your AWS Control Tower landing zone.
- An AFT management account, which you can provision in AWS Control Tower, or provision by other means and enroll into AWS Control Tower.
- A Terraform version and distribution. For more information, see Terraform and AFT versions.
- A VCS provider for tracking and managing changes to code and other files. By default, AFT uses AWS CodeCommit. For more information, see What is AWS CodeCommit? in the *AWS CodeCommit User Guide*.

  If you're deploying AFT for the first time and you don't have an existing CodeCommit repository, you must choose an external VCS provider, such as GitHub or BitBucket. For more information, see Alternatives for version control of source code in AFT.
- A runtime environment where you can run the Terraform module that installs AFT.
- AFT feature options. For more information, see Enable feature options.

Configure and launch your AWS Control Tower Account Factory for Terraform

The following steps assume that you're familiar with the Terraform workflow. You can also learn more about deploying AFT by following the Introduction to AFT lab on the AWS Workshop Studio website.

Step 1: Launch your AWS Control Tower landing zone

Complete the steps in Getting started with AWS Control Tower. This is where you create the AWS Control Tower management account and set up your AWS Control Tower landing zone.

Note

Make sure to create a role for the AWS Control Tower management account that has **AdministratorAccess** credentials. For more information, see the following:

- IAM Identities (users, user groups, and roles) in the *AWS Identity and Access Management User Guide*
- AdministratorAccess in the *AWS Managed Policy Reference Guide*

Step 2: Create a new organizational unit for AFT (strongly recommended)

We recommend that you create a separate OU in your AWS Control Tower landing zone . This OU is where you provision the AFT management account. Create the new OU and the AFT management account from your AWS Control Tower management account. For more information, see Create a new OU.

Step 3: Provision the AFT management account

AFT requires that you provision an AWS account dedicated to AFT management operations. Create the AFT management account when you are signed into the AWS Control Tower management account that is associated to your AWS Control Tower landing zone. You can provision the AFT management account from the AWS Control Tower console by selecting **Create account** on the **Organization** page, or by other means. For more information, see Provision accounts with AWS Service Catalog Account Factory.

Note

If you created a separate OU for AFT, make sure to select this OU when you create the AFT management account.

It can take up to 30 minutes to fully provision the AFT management account.

Step 4: Verify that the Terraform environment is available for deployment

This step assumes that you have experience with Terraform and have procedures in place for executing Terraform. For more information, see [Command: init] on the HashiCorp Developer website.

> **Note**
>
> AFT supports Terraform Version `1.6.0` or later.

Step 5: Optional configurations

- **Optionally set the virtual private cloud (VPC) configuration**

  The AFT module includes a `aft_enable_vpc` parameter, which specifies whether AWS Control Tower provisions account resources within a VPC in the central AFT management account. By default, the parameter is set to `true`. If you set this parameter to `false`, AWS Control Tower deploys AFT *without* the use of a VPC and private networking resources, such as NAT Gateways or VPC endpoints. Disabling `aft_enable_vpc` may help reduce the operating cost of AFT *for some usage patterns*. Adding any VPC configurations overrides the `aft_enable_vpc` parameter being set to `false`.

  > **Note**
  >
  > Re-enabling the `aft_enable_vpc` parameter (switching the value from `false` to `true`) may require you to run the `terraform apply` command twice in succession.

  Instead of provisioning a new VPC, you can configure AFT to use an existing VPC in your account. To use your own VPC, provide the following VPC configuration parameters:

  - `aft_customer_vpc_id` - The ID of your existing VPC
  - `aft_customer_private_subnets` - A list of private subnet IDs in your VPC

  Example configuration:

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory"

  # VPC configuration
```

```
  aft_customer_vpc_id = "vpc-0123456789abcdef0"
  aft_customer_private_subnets = ["subnet-0123456789abcdef0",
 "subnet-0123456789abcdef1"]

  # Other AFT parameters...
}
```

> **Important**
>
> We don't recommend that you use the custom VPC option if you have an existing AFT deployment. You may have dependencies on Lambda functions or CodePipeline that depend on resources within the underlying existing VPC.

- **Optionally configure the Terraform project name**

  You can customize the Terraform project name used by AFT by setting the `terraform_project_name` parameter. By default, AFT puts the deployment in the "default" project in Terraform Cloud or Terraform Enterprise.

  Example configuration:

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory"

  # Project name configuration
  terraform_project_name = "my-organization-aft"

  # Other AFT parameters...
}
```

> **Note**
>
> This parameter is only applicable to Terraform Enterprise or Terraform Cloud deployments.

- **Optionally apply custom tags to AFT resources**

You can apply custom tags to all AFT resources by using the `tags` parameter. These tags help with resource organization, cost allocation, and access control.

Example configuration:

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory"

  # Custom tags configuration
  tags = {
    Environment = "Production"
    CostCenter = "IT-12345"
    Project = "AFT-Deployment"
    Owner = "platform-team@example.com"
  }

  # Other AFT parameters...
}
```

These tags are applied to all resources created by the AFT module. AFT automatically adds a `managed_by = "AFT"` tag to all resources, which can't be overridden by custom tags.

> **Note**
>
> Custom tags can be added at any time, not just during initial deployment.

- **Optionally apply an AWS KMS customer-managed encryption key (CMK) to CloudWatch log groups and SNS topics**

  To enable KMS CMK encryption for log groups and SNS topics, set the `cloudwatch_log_group_enable_cmk_encryption` and `sns_topic_enable_cmk_encryption` variables.

  If you opt into these settings, AFT uses the existing CMK, *alias/aft*, to encrypt CloudWatch logs and SNS topics. This CMK is created when AFT is deployed in the AFT management account, and it can be applied to log groups and SNS topics.

  - If the variable `cloudwatch_log_group_enable_cmk_encryption` is set to **true**, the CloudWatch log groups for AFT are encrypted using the CMK. If the variable is set to **false**, which is the default value, the logs are encrypted using server side encryption with the CloudWatch logs default.
  - If the variable `sns_topic_enable_cmk_encryption` is set to **true**, notifications sent to the AFT SNS topics (*aft-notifications* and *aft-failure-notifications*) are encrypted using the CMK. If the variable is set to **false**, which is the default value, the SNS messages are encrypted with the AWS-managed key: *alias/aws/sns*. For more information, see SSE key terms.

- **Optionally change your CodeBuild compute type**

  During deployment, to change the compute type that AFT uses for CodeBuild, set the variable `aft_codebuild_compute_type`.

  For information about accepted compute types, see About on-demand environment types. The default compute type is `BUILD_GENERAL1_MEDIUM`.

Step 6: Call the Account Factory for Terraform module to deploy AFT

Call the AFT module with the role that you created for the AWS Control Tower management account that has **AdministratorAccess** credentials. AWS Control Tower provisions a Terraform module through the AWS Control Tower management account, which establishes all of the infrastructure required to orchestrate AWS Control Tower Account Factory requests.

You can view the AFT module in the AFT repository on GitHub. The entire GitHub repository is considered the AFT module. Refer to the README file for information about the inputs that are required to run the AFT module and deploy AFT. Alternatively, you can view the AFT module in the Terraform Registry.

If you have pipelines in your environment that are established for managing Terraform, you can integrate the AFT module into your existing workflow. Otherwise, run the AFT module from any environment that's authenticated with the required credentials.

Timeout causes deployment to fail. We recommend using AWS Security Token Service (STS) credentials to ensure you have a timeout that's sufficient for a full deployment. The minimum timeout for AWS STS credentials is 60 minutes. For more information, see Temporary security credentials in IAM in the *AWS Identity and Access Management User Guide*.

> **Note**
>
> You might wait up to 30 minutes for AFT to finish deploying through the Terraform module.

Step 7: Manage the Terraform state file

A Terraform state file is generated when you deploy AFT. This artifact describes the state of the resources that Terraform created. If you plan to update the AFT version, make sure to preseve the Terraform state file, or set up a Terraform backend using Amazon S3 and DynamoDB. The AFT module doesn't manage a backend Terraform state.

> **Note**
>
> You're responsible for protecting the Terraform state file. Some input variables might contain sensitive values, such as a private `ssh` key or Terraform token. Depending on your deployment method, these values can be viewable as plain text in the Terraform state file. For more information, see Sensitive data in State on the HashiCorp website.

Post-deployment steps

After the AFT infrastructure deployment is complete, follow these additional steps to complete the setup process and get ready to provision accounts.

Step 1: Complete CodeConnections with your desired VCS provider

If you choose a third-party VCS provider, AFT establishes CodeConnections, and you confirm them. Refer to Alternatives for version control of source code in AFT to learn how to set up AFT with your preferred VCS.

The initial step of establishing the AWS CodeStar connection is accomplished by AFT. You must confirm the connection.

Step 2: Populate each repository

AFT requires that you manage four repositories:

1. Account requests – This repository handles placing or updating account requests. Examples available . For more information about AFT account requests, see Provision a new account with AFT.
2. AFT account provisioning customizations – This repository manages customizations that are applied to all accounts created by and managed with AFT, before beginning the global customizations stage. Examples available . To create AFT account provisioning customizations, see Create your AFT account provisioning customizations state machine.
3. Global customizations – This repository manages customizations that are applied to all accounts created by and managed with AFT. Examples available . To create AFT global customizations, see Apply global customizations.
4. Account customizations – This repository manages customizations that are applied only to specific accounts created by and managed with AFT. Examples available . To create AFT account customizations, see Apply account customizations.

AFT expects that each of these repositories follow a specific directory structure. The templates that are used to populate your repositories and instructions that describe how to populate the templates are available in the Account Factory for Terraform module in the AFT github repository.

Provision a new account with AFT

This section assumes that you've already set up AFT and your AFT management account, and you are provisioning additional accounts.

To provision a new account with AFT, create an account request Terraform file. This file contains the input for parameters in the **aft-account-request** repository. After creating an account request Terraform file, begin processing your account request by running `git push`. This command invokes the `ct-aft-account-request` operation in the AWS CodePipeline, which is created in the AFT management account after account provisioning finishes. For more information, see AFT account provisioning pipeline.

Account request Terraform file parameters

You must include the following parameters in your account request Terraform file. You can view an example account request Terraform file on GitHub.

- The value of `module name` must be unique per the AWS account request.

- The value of `module source` is the path to the account request Terraform module that AFT provides.
- The value of `control_tower_parameters` captures the required input to create an AWS Control Tower account. The value includes the following input fields:
  - `AccountEmail`
  - `AccountName`
  - `ManagedOrganizationalUnit`
  - `SSOUserEmail`
  - `SSOUserFirstName`
  - `SSOUserLastName`

> **Note**
>
> The input that you provide for `control_tower_parameters` can't be changed during the account provisioning.
> The supported formats for specifying `ManagedOrganizationalUnit` in the **aft-account-request** repository include `OUName` and `OUName (OU-ID)`.

- `account_tags` captures user-defined keys and values, which can tag AWS accounts according to business criteria. For more information, see [Tagging AWS Organizations resources]() in the *AWS Organizations User Guide*.
- The value of `change_management_parameters` captures additional information, such as why an account request was created and who initiated the account request. The value includes the following input fields:
  - `change_reason`
  - `change_requested_by`
- `custom_fields` captures additional metadata with keys and values that deploy as SSM parameters in the vended account under **/aft/account-request/custom-fields/**. You can reference this metadata during account customizations to deploy proper controls. For example, an account that's subject to regulatory compliance might deploy additional AWS Config Rules. The metadata that you collect with `custom_fields` can invoke additional processing during account provisioning and updating. If a custom field is removed from the account request, the custom field is removed from the SSM Parameter Store for the vended account.

- (Optional) `account_customizations_name` captures the account template folder in the **aft-account-customizations** repository. For more information, see [Account customizations].

Submit multiple account requests

AFT processes account requests one at a time, but you can submit multiple account requests to the AFT pipeline. When you submit multiple account requests to the AFT pipeline, AFT queues and processes the account requests in a first-in, first-out order.

> **Note**
>
> You can create an account request Terraform file for each account that you want AFT to provision or cascade multiple account requests in a single account request Terraform file.

Update an existing account

You can update accounts that AFT provisions by editing previously submitted account requests and running `git push`. This command invokes the account provisioning workflow and can process account update requests. You can update the input for `ManagedOrganizationalUnit`, which is part of the required value for `control_tower_parameters`.

`ManagedOrganizationalUnit` is the only parameter that can be updated, among all `control_tower_parameters`. However, other parameters that are part of the account request Terraform file can be updated, such as `custom_fields`. For more information, see [Provision a new account with AFT].

For example, to update the name or email address of an AFT account, you can define the specifics as `custom_fields` in the Account Request file. By doing so, you create SSM parameters, which you can pass into the `aws_account_alternate_contact` resource during global customizations.

```
resource "aws_account_alternate_contact" "operations" {

  alternate_contact_type = "OPERATIONS"

  name          = "Example"
  title         = "Example"
  email_address = "someone@example.com"
```

```
  phone_number   = "+1234567890"
}
```

You can add similar fields for other contact types, such as operations and security. In Global Customizations, add data lookups for each custom field, to ensure that you look up all the fields you created in Account Request:

```
data "aws_ssm_parameter" "billing_name" {
            name = "/aft/account-request/custom-fields/billing_name"
            }

            data "aws_ssm_parameter" "billing_title" {
            name = "/aft/account-request/custom-fields/billing_title"
            }

            data "aws_ssm_parameter" "billing_email_address" {
            name = "/aft/account-request/custom-fields/billing_email_address"
            }

            data "aws_ssm_parameter" "billing_phone_number" {
            name = "/aft/account-request/custom-fields/billing_phone_number"
            }
```

Finally, also in the Global Customizations file, create the alternate contact resources. You will need to define one of these blocks for every contact type you created in Account Request:

```
resource "aws_account_alternate_contact" "billing" {

            alternate_contact_type = "BILLING"

            name          = data.aws_ssm_parameter.billing_name.value
            title         = data.aws_ssm_parameter.billing_title.value
            email_address = data.aws_ssm_parameter.billing_email_address.value
            phone_number  = data.aws_ssm_parameter.billing_phone_number.value
            }
```

Note

The input that you provide for `control_tower_parameters` can't be changed during account provisioning.

The supported formats for specifying `ManagedOrganizationalUnit` in the **aft-account-request** repository include `OUName` and `OUName (OU-ID)`.

Update an account that AFT doesn't provision

You can update AWS Control Tower accounts created outside of AFT by specifying the account in the **aft-account-request** repository.

Note

Make sure that all account details are correct and consistent with the AWS Control Tower organization and respective AWS Service Catalog provisioned product.

Prerequisites for updating an existing AWS account with AFT

- The AWS account must be enrolled in AWS Control Tower.
- The AWS account must be part of the AWS Control Tower organization.

Terraform and AFT versions

Account Factory for Terraform (AFT) supports Terraform version `1.6.0` or later. You must provide a Terraform version as an input parameter for the AFT deployment process, as shown in the example that follows.

```
terraform_version = "1.6.0"
```

Terraform distributions

AFT supports three Terraform distributions:

- Terraform Community Edition
- Terraform Cloud
- Terraform Enterprise

These distributions are explained in the sections that follow. Provide the Terraform distribution of your choice as an input parameter during the AFT bootstrap process. For more information on AFT deployment and input parameters, see Deploy AWS Control Tower Account Factory for Terraform (AFT) .

If you choose the Terraform Cloud or Terraform Enterprise distributions, the API token you specify for `terraform_token` must be a User or Team API token. An Organization token is not supported for all required APIs. For security reasons, you must avoid checking in this token's value to your version control system (VCS) by assigning a terraform variable, as shown in the example that follows.

```
# Sensitive variable managed in Terraform Cloud:
terraform_token = var.terraform_cloud_token
```

Terraform Community Edition

When you select Terraform Community Edition as your distribution, AFT manages the Terraform backend for you in the AFT management account. AFT downloads the `terraform-cli` of your specified Terraform version to run during the AFT deployment and the AFT pipeline phases. The resulting Terraform state configuration is stored in an Amazon S3 bucket, named with the following form:

```
aft-backend-[account_id]-primary-region
```

AFT also creates an Amazon S3 bucket that replicates your Terraform state configuration in another AWS Region, for disaster recovery purposes, named with the following form:

```
aft-backend-[account_id]-secondary-region
```

We recommend that you enable multi-factor authentication (MFA) for delete functions on these Terraform state Amazon S3 buckets. To learn more about Terraform Community Edition, see the Terraform documentation .

To select Terraform OSS as your distribution, provide the following input parameter:

```
terraform_distribution = "oss"
```

Terraform Cloud

When you select Terraform Cloud as your distribution, AFT creates workspaces for the following components in your Terraform Cloud organization, which initiates an API-driven workflow.

- Account request
- AFT customizations for accounts that AFT provisions
- Account customizations for accounts that AFT provisions
- Global customizations for accounts that AFT provisions

Terraform Cloud manages the resulting Terraform state configuration.

When you select Terraform Cloud as your distribution, provide the following input parameters:

- `terraform_distribution = "tfc"`
- `terraform_token` – This parameter contains the value of the Terraform Cloud token. AFT marks the as sensitive and stores the value as a secure string in the SSM parameter store in the AFT management account. We recommend that you periodically rotate the value of the Terraform token according to your company's security policies and compliance guidelines. The Terraform token should be a User or Team level API token. Organization tokens are not supported.
- `terraform_org_name` – This parameter contains the name of your Terraform Cloud organization.

> **Note**
>
> Multiple AFT deployments in a single Terraform Cloud organization is not supported.

For information about how to set up Terraform Cloud, see the Terraform documentation.

Terraform Enterprise

When you select Terraform Enterprise as your distribution, AFT creates workspaces for the following components in your Terraform Enterprise organization, and it triggers API-driven workflow for the resulting Terraform runs.

- Account request
- AFT account provisioning customizations for accounts provisioned by AFT
- Account customizations for accounts provisioned by AFT
- Global customizations for accounts provisioned by AFT

The resulting Terraform state configuration is managed by your Terraform Enterprise setup.

To select Terraform Enterprise as your distribution, provide the following input parameters:

- `terraform_distribution = "tfe"`
- `terraform_token` – This parameter contains the value of your Terraform Enterprise token. AFT marks its value as sensitive and stores it as a secure string in the SSM parameter store, in the AFT management account. We recommend that you periodically rotate the value of the Terraform token, according to your company's security policies and compliance guidelines.
- `terraform_org_name` – This parameter contains the name of your Terraform Enterprise organization.
- `terraform_api_endpoint` – This parameter contains the URL of your Terraform Enterprise environment. The value of this parameter must be in the format:

```
https://{fqdn}/api/v2/
```

See the Terraform documentation to learn more about how to set up Terraform Enterprise.

Check the AFT version

You can check your deployed AFT version by querying the AWS SSM Parameter Store key:

```
/aft/config/aft/version
```

If you use the registry method, you can pin the version.

```
module "control_tower_account_factory" {
  source  = "aws-ia/control_tower_account_factory/aws"
  version = "1.3.2"
  # insert the 6 required variables here
}
```

You can view more information about AFT versions in the AFT repository.

Update the AFT version

Sign in to the AWS Control Tower management account to initiate this AFT update.

You can update your deployed AFT version by pulling it in from the `main` repository branch:

```
terraform get -update
```

After the pull is complete, you can re-run the Terraform plan or run apply to update the AFT infrastructure with the latest changes.

Enable feature options

AFT offers feature options based on best practices. You can opt-in to these features, by means of feature flags, during AFT deployment. Refer to Provision a new account with AFT for more information about AFT input configuration parameters.

These features are not enabled by default. You must explicitly enable each one in your environment.

Topics

- AWS CloudTrail data events
- AWS Enterprise Support plan
- Delete the AWS default VPC

AWS CloudTrail data events

When enabled, the AWS CloudTrail data events option configures these capabilities.

- Creates an Organization Trail in the AWS Control Tower management account, for CloudTrail
- Turns on logging for Amazon S3 and Lambda data events
- Encrypts and exports all the CloudTrail data events to an `aws-aft-logs-*` S3 bucket in the AWS Control Tower Log Archive account, with AWS KMS encryption
- Turns on the **Log file validation** setting

To enable this option, set the following feature flag to **True** in your AFT deployment input configuration.

```
aft_feature_cloudtrail_data_events
```

Prerequisite

Before you enable this feature option, be sure that trusted access for AWS CloudTrail is enabled in your organization.

To check the status of trusted access for CloudTrail :

1. Navigate to the AWS Organizations console.
2. Choose **Services > CloudTrail**.
3. Then select **Enable trusted access** in the upper right, if needed.

You may receive a warning message that advises you to use the AWS CloudTrail console, but in this case, disregard the warning. AFT creates the trail as part of enabling this feature option, after you allow trusted access. If trusted access is not enabled, you will receive an error message when AFT attempts to create your trail for data events.

> **Note**
>
> This setting works at the organization level. Enabling this setting affects all accounts in AWS Organizations, whether they are managed by AFT or not. All buckets in the AWS Control Tower Log Archive account at the time of enabling are excluded from Amazon S3 data events. Refer to the AWS CloudTrail User Guide to learn more about CloudTrail.

AWS Enterprise Support plan

When this option is enabled, the AFT pipeline turns on the AWS Enterprise Support plan for accounts provisioned by AFT.

AWS accounts by default come with the AWS Basic Support plan enabled. AFT provides automated enrollment into the enterprise support level, for accounts that AFT provisions. The provisioning process opens a support ticket for the account, requesting it to be added to the AWS Enterprise Support plan.

To enable the Enterprise Support option, set the following feature flag to **True** in your AFT deployment input configuration.

```
aft_feature_enterprise_support=false
```

Refer to Compare AWS Support Plans to learn more about AWS Support Plans.

> **Note**
>
> To allow this feature to operate, you must enroll the payer account into the Enterprise Support plan.

Delete the AWS default VPC

When you enable this option, AFT deletes all AWS default VPCs in the AFT management account and in all AWS Regions, even if haven't deployed AWS Control Tower resources in those AWS Regions.

AFT doesn't delete AWS default VPCs automatically for any AWS Control Tower accounts that AFT provisions or for existing AWS accounts that you enroll in AWS Control Tower through AFT.

New AWS accounts are created with a VPC set up in each AWS Region, by default. Your enterprise may have standard practices for creating VPCs, which require you to delete the AWS default VPC and avoid enabling it, especially for the AFT management account.

To enable this option, set the following feature flag to **True** in your AFT deployment input configuration.

```
aft_feature_delete_default_vpcs_enabled
```

The following is an example of a AFT deployment input configuration.

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory"
  ct_management_account_id    = var.ct_management_account_id
  log_archive_account_id      = var.log_archive_account_id
  audit_account_id            = var.audit_account_id
  aft_management_account_id   = var.aft_management_account_id
  ct_home_region              = var.ct_home_region
  tf_backend_secondary_region = var.tf_backend_secondary_region

  vcs_provider                                  = "github"
  account_request_repo_name                     = "${var.github_username}/learn-
terraform-aft-account-request"
  account_provisioning_customizations_repo_name = "${var.github_username}/learn-
terraform-aft-account-provisioning-customizations"
```

```
  global_customizations_repo_name               = "${var.github_username}/learn-
terraform-aft-global-customizations"
  account_customizations_repo_name              = "${var.github_username}/learn-
terraform-aft-account-customizations"

  # Optional Feature Flags
  aft_feature_delete_default_vpcs_enabled = true
  aft_feature_cloudtrail_data_events      = false
  aft_feature_enterprise_support          = false
}
```

Refer to Default VPC and default subnets to learn more about default VPCs.

Resource considerations for AWS Control Tower Account Factory for Terraform

When you set up your landing zone using AWS Control Tower Account Factory for Terraform, several types of AWS resources are created within your AWS accounts.

Search for resources

- You can use tags to search for the most updated list of AFT resources. The key-value pair for your search is:

  ```
  Key: managed_by | Value: AFT
  ```

- For component services that do not support tags, you can locate resources with a search for `aft` in the resource names.

> **Note**
>
> AFT does not create any AWS Backup resources in the management account.

Tables of resources initially created, by account

AWS Control Tower Account Factory for Terraform management account

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	AWSAFTAdmin AWSAFTExecution AWSAFTService ct-aft-* aft-* codebuild_trigger_role python-layer-builder-aft-common-*
AWS Identity and Access Management	Policies	aft-*
CodeCommit	Repositories	aft-*
CodeBuild	Build Projects	aft-* ct-aft-* python-layer-builder-aft-common-*
Code Pipeline	Pipelines	*YourAccountId* -customizations-pipeline
Amazon S3	Buckets	aft-*
Lambda	Functions	aft-*
Lambda	Layers	aft-common-*
DynamoDB	Tables	aft-request aft-request-audit

AWS service	Resource type	Resource name
		aft-request-metadata aft-controltower-events
Step Functions	State Machines	aft-account-provisioning-cu stomizations aft-account-provisioning-fr amework aft-feature-options aft-invoke-customizations
VPC	VPC	aft-management-vpc
Amazon SNS	Topics	aft-notifications aft-failure-notifications
Amazon EventBridge	Event buses	aft-events-from-ct-management
Amazon EventBridge	Event rules	aft-account-provisioning-cu stomizations-trigger aft-account-request-codepip eline-trigger aft-lambda-account-request-processor aft-controltower-event-logg er
Key Management Service (KMS)	Customer Managed Keys	aft-backend-*-kms-key aft
AWS Systems Manager	Parameter store	/aft/*

AWS service	Resource type	Resource name
Amazon SQS	Queues	aft-account-request.fifo aft-account-request-dlg.fifo
CloudWatch	Log groups	/aws/*/ct-aft-* /aws/*/aft-* /aws/codebuild/python-layer-builder-aft-common-*
AWS Backup	Vaults	aft-controltower-backup-vault
AWS Backup	Plans	aft-controltower-backup-plan
AWS Support Center (Optional)	Support plans	Enterprise

AWS accounts provisioned through AWS Control Tower Account Factory for Terraform

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	AWSAFTExecution
AWS Support Center (Optional)	Support plans	Enterprise

AWS Control Tower management account

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	AWSAFTExecution AWSAFTService

AWS service	Resource type	Resource name
		aft-controltower-events-rule
AWS Systems Manager	Parameter store	/aft/*
EventBridge	Event rules	aft-capture-ct-events
CloudTrail (Optional)	Trails	aws-aft-CustomizationsCloud Trail
AWS Support Center (Optional)	Support plans	Enterprise

AWS Control Tower log archive account

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	AWSAFTExecution AWSAFTService
Key Management Service (KMS)	Customer Managed Keys	aft
Amazon S3	Buckets	aws-aft-logs-* aws-aft-s3-access-logs-*
AWS Support Center (Optional)	Support plans	Enterprise

AWS Control Tower audit account

AWS service	Resource type	Resource name
AWS Identity and Access Management	Roles	AWSAFTExecution AWSAFTService

AWS service	Resource type	Resource name
AWS Support Center (Optional)	Support plans	Enterprise

Required roles

In general, roles and policies are part of identity and access management (IAM) in AWS. Refer to the *AWS IAM User Guide* for more information.

AFT creates multiple IAM roles and policies in the AFT management and AWS Control Tower management accounts to support the operations of the AFT pipeline. These roles are created based on the least privilege access model, which restricts permission to the minimally required sets of actions and resources for each role and policy. These roles and policies are assigned an AWS tag `key:value` pair, as `managed_by:AFT` for identification.

Besides these IAM roles, AFT creates three essential roles:

- the `AWSAFTAdmin` role
- the `AWSAFTExecution` role
- the `AWSAFTService` role

These roles are explained in the following sections.

The AWSAFTAdmin role, explained

When you deploy AFT, the `AWSAFTAdmin` role is created in the AFT management account. This role allows the AFT pipeline to assume the `AWSAFTExecution` role in AWS Control Tower and AFT provisioned accounts, thereby to perform actions related to account provisioning and customizations.

Here is the inline policy (JSON artifact) attached to the `AWSAFTAdmin` role:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::*:role/AWSAFTExecution",
                "arn:aws:iam::*:role/AWSAFTService"
            ]
        }
    ]
}
```

The following JSON artifact shows the trust relationship for the `AWSAFTAdmin` role. The placeholder number `012345678901` is replaced by the AFT management account ID number.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The AWSAFTExecution role, explained

When you deploy AFT, the `AWSAFTExecution` role is created in the AFT management and AWS Control Tower management accounts. Later, the AFT pipeline creates the `AWSAFTExecution` role in each AFT provisioned account during the AFT account provisioning stage.

AFT utilizes the `AWSControlTowerExecution` role initially, to create the `AWSAFTExecution` role in specified accounts. The `AWSAFTExecution` role allows the AFT pipeline to run the steps that are performed during the AFT framework's provisioning and provisioning customizations stages, for AFT provisioned accounts and for shared accounts.

Distinct roles help you limit scope

As a best practice, keep the customization permissions separate from the permissions allowed during your initial deployment of resources. Remember that the `AWSAFTService` role is intended for account provisioning, and the `AWSAFTExecution` role is intended for account customization. This separation limits the scope of permissions that are allowed during each phase of the pipeline. This distinction is especially important if you are customizing the AWS Control Tower shared accounts, because the shared accounts may contain sensitive information, such as billing details or user information.

Permissions for `AWSAFTExecution` role: **AdministratorAccess** – an AWS managed policy

The following JSON artifact shows the IAM policy (trust relationship) attached to the `AWSAFTExecution` role. The placeholder number `012345678901` is replaced by the AFT management account ID number.

Trust policy for `AWSAFTExecution`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/AWSAFTAdmin"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The AWSAFTService role, explained

The `AWSAFTService` role deploys AFT resources in all enrolled and managed accounts, including the shared accounts and management account. Resources formerly were deployed by the `AWSAFTExecution` role only.

The `AWSAFTService` role is intended for use by the service infrastructure to deploy resources during the provisioning stage, and the `AWSAFTExecution` role is intended to be used only to deploy customizations. By assuming the roles in this way, you can maintain more granular access control during the each stage.

Permissions for `AWSAFTService` role: **AdministratorAccess** – an AWS managed policy

The following JSON artifact shows the IAM policy (trust relationship) attached to the `AWSAFTService` role. The placeholder number `012345678901` is replaced by the AFT management account ID number.

Trust policy for `AWSAFTService`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/AWSAFTAdmin"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Component services

When you deploy AFT, components are added to your AWS environment from each of these AWS services.

- **AWS Control Tower** – AFT uses AWS Control Tower Account Factory in the AWS Control Tower management account to provision accounts.
- **Amazon DynamoDB** – AFT creates Amazon DynamoDB tables in the AFT management account, which store account requests, audit history of account updates, account metadata, and AWS Control Tower lifecycle events. AFT also creates DynamoDB Lambda triggers to initiate downstream processes, such as starting the AFT account provisioning workflow.

- **Amazon Simple Storage Service** – AFT creates Amazon Simple Storage Service (S3) buckets in the AFT management account and the AWS Control Tower log archive account, which store logs generated by the AWS services that the AFT pipeline requires. AFT also creates a Terraform backend S3 bucket, in primary and secondary AWS Regions, to store Terraform states generated during AFT pipeline workflows.
- **Amazon Simple Notification Service** – AFT creates Amazon Simple Notification Service (SNS) topics in the AFT management account, which stores success and failure notifications after processing every AFT account request. You may receive these messages using your choice of protocol.
- **Amazon Simple Queuing Service** – AFT creates an Amazon Simple Queuing Service (Amazon SQS) FIFO queue in the AFT management account. The queue allows you to submit multiple account requests in parallel, but it sends one request at a time to AWS Control Tower Account Factory, for sequential processing.
- **AWS CodeBuild** – AFT creates AWS CodeBuild build projects in the AFT management account to initialize, compile, test, and apply Terraform plans for AFT source code in various build stages.
- **AWS CodePipeline** – AFT creates AWS CodePipeline pipelines in the AFT management account to integrate with your selected, supported AWS CodeStar connections provider for AFT source code, and to trigger build jobs in AWS CodeBuild.
- **AWS Lambda** – AFT creates AWS Lambda functions and layers in the AFT management account to perform steps during the account request, AFT account provisioning, and account customizations processes.
- **AWS Systems Manager Parameter Store** – AFT sets up the AWS Systems Manager Parameter Store in the AFT management account, to store the configuration parameters required for the AFT pipeline processes.
- **Amazon CloudWatch** – AFT creates Amazon CloudWatch log groups in the AFT management account to store logs generated by AWS services employed by the AFT pipeline. The retention period for CloudWatch logs is set to `Never Expire`.
- **Amazon VPC** – AFT creates an Amazon Virtual Private Cloud (VPC) to isolate services and resources in the AFT management account into a separate networking environment, for enhanced security.
- **AWS KMS** – AFT uses the AWS Key Management Service (KMS) in the AFT management account and in the AWS Control Tower log archive account. AFT creates keys to encrypt Terraform states, data stored in DynamoDB tables, and SNS topics. These logs and artifacts are generated when AWS resources and services are deployed by AFT. KMS keys created by AFT have yearly rotation enabled by default.

- **AWS Identity and Access Management (IAM)** – AFT follows the recommended Least Privilege model. It creates AWS Identity and Access Management (IAM) roles and policies in the AFT management account, in AWS Control Tower accounts, and in AFT provisioned accounts, as needed, to perform actions required during the AFT pipeline workflow.
- **AWS Step Functions** – AFT creates AWS Step Functions state machines in the AFT management account. These state machines orchestrate and automate the process and steps for the AFT account provisioning framework and customizations.
- **Amazon EventBridge** – AFT creates an Amazon EventBridge event bus in the AFT and AWS Control Tower management account to capture and store AWS Control Tower lifecycle events long-term in the AFT management account's DynamoDB table. AFT creates Amazon CloudWatch event rules in the AFT management and AWS Control Tower management accounts, which trigger multiple steps required during running of the AFT pipeline workflow
- **AWS CloudTrail (Optional)** – When this feature is enabled, AFT creates an AWS CloudTrail organization trail in the AWS Control Tower management account, for logging data events for Amazon S3 buckets and AWS Lambda functions. AFT sends these logs to a central S3 bucket in the AWS Control Tower log archive account.
- **AWS Support (Optional)** – When this feature is enabled, AFT turns on the AWS Enterprise Support plan for accounts provisioned by AFT. By default, AWS accounts are created with the AWS Basic Support plan enabled.

AFT account provisioning pipeline

After the account provisioning stage of the pipeline is complete, the AFT framework continues. It automatically runs a series of steps to ensure that the newly provisioned accounts have details in place, before the Account customizations stage begins.

Here are the next steps that the AFT pipeline runs.

1. Validates the account request input.
2. Retrieves information about the account provisioned, for example, the account ID.
3. Stores the account metadata in a DynamoDB table in the AFT management account.
4. Creates the **AWSAFTExecution** IAM role in the newly provisioned account. AFT assumes this role to perform the account customizations stage, because this role grants access to the account factory portfolio.
5. Applies the account tags that you provided as part of the account request input parameters.

6. Applies the AFT feature options you chose at the time of AFT deployment.
7. Applies the AFT account provisioning customizations you provided. The next section tells more about how to set up these customizations with an AWS Step Functions state machine, in a `git` repository. This stage is sometimes referred to as the *account provisioning framework* stage. It is part of the core provisioning process, but you've previously set up a framework that delivers customized integrations as part of your account provisioning workflow, before additional customizations are added to the accounts in the next stage.
8. For each account provisioned, it creates an AWS CodePipeline in the AFT management account, which will run to perform the (next, global) [Account customizations](#) stage.
9. Invokes the account customizations pipeline for each account provisioned (and targeted).
10. Sends a success or failure notification to the SNS topic, from which you can retrieve the messages.

Set up the account provisioning framework customizations with a state machine

If you set up custom, non-Terraform integrations before you provision your accounts, these customizations are included in your AFT account provisioning workflow. For example, you may require certain customizations to ensure that all accounts created by AFT are compliant with the standards and policies of your organization, such as security standards, and these standards may be added to accounts before additional customization. These *account provisioning framework* customizations are implemented on every provisioned account, before the global account customization stage begins next.

> **Note**
>
> The AFT feature described in this section is intended for advanced users who understand the functioning of AWS Step Functions. As an alternative, we recommend that you work with the global helpers in the account customizations stage.

The AFT account provisioning framework calls an AWS Step Functions state machine, which you define, to implement your customizations. Refer to the [AWS Step Functions documentation](#) to learn more about the possible state machine integrations.

Here are some common integrations.

- AWS Lambda functions in the language of your choice

- AWS ECS or AWS Fargate tasks, using Docker containers
- AWS Step Functions activities using custom workers, hosted either in AWS or on-premises
- Amazon SNS or SQS integrations

If no AWS Step Functions state machine is defined, the stage passes with a no-op. To create an AFT account provisioning customizations state machine, follow the instructions in Create your AFT account provisioning customizations state machine. Before you add customizations, be sure you have the prerequisites in place.

These types of integrations are not part of AWS Control Tower, and they cannot be added during the global pre-API stage of AFT account customization. Instead, the AFT pipeline allows you to set up these customizations as part of the provisioning process, and they are run in the provisioning workflow. You must implement these customizations by creating your state machine ahead of time, before you kick off the AFT account provisioning stage, as described in the following sections.

Prerequisites for creating a state machine

- A fully deployed AFT. See Deploy AWS Control Tower Account Factory for Terraform (AFT) for more information about AFT deployment.
- Set up a `git` repository in your environment for AFT account provisioning customizations. See Post-deployment steps for more information.

Create your AFT account provisioning customizations state machine

Step 1: Modify the state machine definition

Modify the example `customizations.asl.json` state machine definition. The example is available in the `git` repository you set up for storing AFT account provisioning customizations, in your post-deployment steps. Refer to the AWS Step Functions Developer Guide to learn more about state machine definitions.

Step 2: Include the corresponding Terraform configuration

Include Terraform files with the `.tf` extension in the same `git` repository with the state machine definition for your custom integration. For example, if you choose to call a Lambda function in your state machine task definition, you'd include the `lambda.tf` file in the same directory. Make sure you include the required IAM roles and permissions for your custom configurations.

When you provide the appropriate input, the AFT pipeline automatically invokes your state machine and deploys your customizations as part of the AFT account provisioning framework stage.

To re-start the AFT account provisioning framework and customizations

AFT runs the account provisioning framework and customizations steps for every account vended through the AFT pipeline. To re-start account provisioning customizations, you can use one of these two methods:

1. Make any change to an existing account in the account request repo.
2. Provision a new account with AFT.

Account customizations

AFT can deploy standard or customized configurations in provisioned accounts. In the AFT management account, AFT provides one pipeline for each account. With this pipeline, you can implement your customizations in all accounts, in a set of accounts, or in individual accounts. You can run Python scripts, bash scripts, and Terraform configurations, or you can interact with the AWS CLI as part of your account customizations stage.

Overview

After your customizations are specified in your chosen `git` repositories, either the one where you store your global customizations or where you store your account customizations, the account customizations stage is completed automatically by the AFT pipeline. To customize accounts retroactively, see Re-invoke customizations.

Global customizations (optional)

You can choose to apply certain customizations to all accounts that are provisioned by AFT. For example, if you need to create a particular IAM role, or to deploy a custom control in every account, the global customizations stage in AFT pipeline allows you to do so, automatically.

Account customizations (optional)

To customize an individual account, or a set of accounts, differently than other AFT provisioned accounts, you can leverage the account customizations portion of the AFT pipeline to implement account-specific configurations. For example, only a certain account may require access to an internet gateway.

Customization prerequisites

Before you begin to customize accounts, be sure these prerequisites are in place.

- A fully deployed AFT. For information about how to deploy, see Configure and launch your AWS Control Tower Account Factory for Terraform.
- Pre-populated `git` repositories for global customizations and account customizations in your environment. See *Step 3: Populate each repository* in Post-deployment steps for more information.

Apply global customizations

To apply global customizations, you must push a specific folder structure to your chosen repository.

- If your custom configurations are in the form of Python programs or scripts, place those under **api_helpers/python** folder in your repository.
- If your custom configurations are in the form of Bash scripts, place those under **api_helpers** folder in your repository.
- If your custom configurations are in the form of Terraform, place those under the **terraform** folder in your repository.
- Refer to the global customizations README file for more details on creating custom configurations.

Note

Global customizations are applied automatically, after the AFT account provisioning framework stage in the AFT pipeline.

Apply account customizations

You can apply account customizations by pushing a specific folder structure to your chosen repository. Account customizations are applied automatically in the AFT pipeline and after the global customizations stage. You can also create multiple folders that contain different account customizations in your account customizations repository. For each account customization that you require, use the following steps.

To apply account customizations

1. **Step 1: Create a folder for an account customization**

   In your chosen repository, copy the `ACCOUNT_TEMPLATE` folder that AFT provides to a new folder. The name of your new folder should match the `account_customizations_name` that you provide in your account request.

2. **Add the configurations to your specific account customizations folder**

   You can add configurations to your account customizations folder based on the format of your configurations.

   - If your custom configurations are in the form of Python programs or scripts, place them under the ***[account_customizations_name]*/api_helpers/python** folder that's in your repository.
   - If your custom configurations are in the form of Bash scripts, place them under the ***[account_customizations_name]*/api_helpers** folder that's in your repository.
   - If your custom configurations are in the form of Terraform, place them under the ***[account_customizations_name]*/terraform** folder that's in your repository.

   For more information about creating custom configurations, refer to the account customizations README file.

3. **Refer to the specific `account_customizations_name` parameter in the account request file**

   The AFT account request file includes the input parameter `account_customizations_name`. Enter the name of your account customization as the value for this parameter.

> **Note**
>
> You can submit multiple account requests for accounts in your environment. When you want to apply different or similar account customizations, specifiy the account

customizations using the `account_customizations_name` input parameter in your account requests. For more information, see Submit multiple account requests.

Re-invoke customizations

AFT provides a way to re-invoke customizations in the AFT pipeline. This method is useful when you've added a new customization step, or when you are making changes to an existing customization. When you re-invoke, AFT initiates the customizations pipeline to make changes to the AFT provisioned account. An event-source-based re-invoke allows you to apply customizations to individual accounts, to all accounts, to accounts according to their OU, or to accounts selected according to tags.

Follow these three steps to re-invoke customizations for AFT-provisioned accounts.

Step 1: Push changes to global or account customizations `git` repositories

You can update your global and account customizations as needed and push changes back to your `git` repositories. At this point, nothing happens, The customizations pipeline must be invoked by an event source, as explained in the next two steps.

Step 2: Start an AWS Step Function run for re-invoking customizations

AFT provides an AWS Step Function called `aft-invoke-customizations` in the AFT management account. The purpose of that function is to re-invoke the customization pipeline for AFT-provisioned accounts.

Here is an example of an event schema (JSON format) you can create to pass input to the `aft-invoke-customizations` AWS Step Function.

```
{
  "include": [
    {
      "type": "all"
    },
    {
      "type": "ous",
      "target_value": [ "ou1","ou2"]
    },
    {
      "type": "tags",
```

```
      "target_value": [ {"key1": "value1"}, {"key2": "value2"}]
    },
    {
      "type": "accounts",
      "target_value": [ "acc1_ID","acc2_ID"]
    }
  ],

  "exclude": [
    {
      "type": "ous",
      "target_value": [ "ou1","ou2"]
    },
    {
      "type": "tags",
      "target_value": [ {"key1": "value1"}, {"key2": "value2"}]
    },
    {
      "type": "accounts",
      "target_value": [ "acc1_ID","acc2_ID"]
    }
  ]
}
```

The example event schema shows that you can choose accounts to include or exclude from the re-invoke process. You can filter by organizational unit (OU), account tags, and account ID. If you don't apply any filters and include the statement `"type":"all"`, the customization for all AFT-provisioned accounts is re-invoked.

> **Note**
>
> If your version of AWS Control Tower Account Factory for Terraform (AFT) is 1.6.5 or later, you can target nested OUs with the syntax `OU Name (ou-id-1234)`. For more information, see the following topic on [GitHub](https://github.com).

After you fill out the event parameters, Step Functions runs and invokes the corresponding customizations. AFT can invoke a maximum of 5 customizations at a time. Step Functions waits and loops until all accounts matching the event criteria are complete.

Step 3: Monitor the AWS Step Function output and watch AWS CodePipeline running

- The resulting Step Function output contains account IDs that match the Step Function input event source.
- Navigate to AWS CodePipeline under **Developer Tools** and view the corresponding customization pipelines for the account ID.

Troubleshooting with AFT account customization request tracing

Account customization workflows that are based on AWS Lambda emit logs containing target account and customization request IDs. AFT allows you to trace and troubleshoot customization requests with Amazon CloudWatch Logs by providing you with CloudWatch Logs Insights queries that you can use to filter CloudWatch Logs related to your customization request by your target account or customization request ID. For more information, see [Analyzing log data with Amazon CloudWatch Logs](#) in the *Amazon CloudWatch Logs User Guide*.

To use CloudWatch Logs Insights for AFT

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. From the navigation pane, choose **Logs**, and then choose **Logs insights**.
3. Choose **Queries**.
4. Under **Sample queries**, choose **Account Factory for Terraform**, and then select one of the following queries:

   - **Customization Logs by Account ID**

     > **Note**
     >
     > Make sure to replace *"YOUR-ACCOUNT-ID"* with your target account ID.

     ```
     fields @timestamp, log_message.account_id as target_account_id,
      log_message.customization_request_id as customization_request_id,
      log_message.detail as detail, @logStream
     | sort @timestamp desc
     | filter log_message.account_id == "YOUR-ACCOUNT-ID" and @message like /
     customization_request_id/
     ```

   - **Customization Logs by Customization Request ID**

> **Note**
>
> Make sure to replace *"YOUR-CUSTOMIZATION-REQUEST-ID"* with your customization request ID. You can find your customization request ID in the output of the AFT account provisioning framework AWS Step Functions state machine. For more information about the AFT account provisioning framework, see AFT account provisioning pipeline

```
fields @timestamp, log_message.account_id as target_account_id,
 log_message.customization_request_id as customization_request_id,
 log_message.detail as detail, @logStream
| sort @timestamp desc
| filter log_message.customization_request_id == "YOUR-CUSTOMIZATION-REQUEST-ID"
```

5. After you select a query, make sure to select a time interval, and then choose **Run query**.

Alternatives for version control of source code in AFT

AFT uses AWS CodeCommit for a source code version control system (VCS), and it allows other CodeConnections that meet your business requirements or existing architecture.

If you're deploying AFT for the first time and you don't have an existing CodeCommit repository, you must specify an external VCS provider, as part of the AFT deployment prerequisites.

AFT supports the following source code control alternatives:

- GitHub
- GitHub Enterprise Server
- BitBucket
- GitLab
- GitLab Self-managed

> **Note**
>
> If you specify AWS CodeCommit as your VCS, no additional steps are required. AFT creates the necessary `git` repositories in your environment, with default names. However, you can override the default repository names for CodeCommit, as needed, to comply with your organizational standards.

Set up an alternative source code version control system (custom VCS) with AFT

To set up an alternative source code version control system for your AFT deployment, follow these steps.

Step 1: Create `git` repositories in a supported third-party version control system (VCS).

If you are not using AWS CodeCommit, you must create `git` repositories in your AFT-supported, third-party VCS provider environment for the following items.

- **AFT account requests.** Sample code available . For more information about AFT account requests, see Provision a new account with AFT.
- **AFT account provisioning customizations.** Sample code available . For more information on AFT account provisioning customizations, see Create your AFT account provisioning customizations state machine.
- **AFT global customizations.** Sample code available . For more information on AFT global customizations, see Account customizations.
- **AFT account customizations.** Sample code available . For more information on AFT account customizations, see Account customizations.

Step 2: Specify the VCS configuration parameters required for AFT deployment

The following input parameters are needed to configure your VCS provider as part of the AFT deployment.

- **vcs_provider**: If you are not using AWS CodeCommit, specify the VCS provider as `"bitbucket"`, `"github"`, `"githubenterprise"`, or `"gitlab"`, based on your use case.
- **github_enterprise_url**: For GitHub Enterprise customers only, specify the GitHub URL.
- **account_request_repo_name**: For AWS CodeCommit users, this value is set to `aft-account-request`. In an AFT-supported, third-party VCS provider environment, update this input value

with your actual repository name. For BitBucket, Github, GitHub Enterprise, GitLab, and GitLab Self-managed, the repository name must have the format `[Org]/[Repo]`.

- **account_customizations_repo_name**: For AWS CodeCommit users, this value is set to `aft-account-customizations`. In an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, GitHub Enterprise, GitLab, and GitLab Self-managed, the repository name must have the format `[Org]/[Repo]`.
- **account_provisioning_customizations_repo_name**: For AWS CodeCommit users, this value is set to `aft-account-provisioning-customizations`. In an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, GitHub Enterprise, GitLab, and GitLab Self-managed, the repository name must have the format `[Org]/[Repo]`.
- **global_customizations_repo_name**: For AWS CodeCommit users, this value is set to `aft-global-customizations`. In an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, GitHub Enterprise, GitLab, and GitLab Self-managed, the repository name must have the format `[Org]/[Repo]`.
- **account_request_repo_branch**: The branch is `main` by default, but the value can be overridden.

By default, AFT sources from the `main` branch of each `git` repository. You can override the branch name value with an additional input parameter. For more information about input parameters, refer to the README file in the AFT Terraform module.

> **For existing AWS CodeCommit customers**
>
> If you create a CodeCommit repository with a new name for AFT, you can update the repository name by updating the values for these input parameters.

Step 3: Complete the AWS CodeCommit connection for third-party VCS providers

When your deployment runs, AFT either creates the required AWS CodeCommit repositories, or it creates an AWS CodeCommit connection for your chosen third-party VCS provider. In case of the latter, you must manually sign in to the AFT management account's console to complete the pending CodeCommit connection. See the AWS CodeCommit documentation for further instructions on completing the CodeCommit connection.

Move AFT from AWS CodeCommit to another VCS provider

This section provides an overview of how you can move AWS Control Tower Account Factory for Terraform (AFT) from AWS CodeCommit as your version control system (VCS) to another VCS provider.

Step 1. Set up new repositories in the VCS of your choice.

Step 2. Add these repositories as new remotes in `git`.

Step 3. Execute `git push` to the new VCS provider.

> **Note**
>
> The repository structure that you create should be the same as in AWS CodeCommit. Changing the structure impedes the ability of AFT to execute the desired code.
>
> **Repository structure:**
>
> - aft-account-request
> - aft-account-customizations
> - aft-global-customizations
> - aft-account-provisioning-customizations

Step 4. In your AWS Control Tower management account, update the Terraform module (bootstrap) to point to your VCS provider, as shown in the following example:

Example: GitLab with Terraform OSS

– Perform `terraform plan` to preview changes, then `terraform apply`.

Step 5. Complete the steps to finish setting up the CodeConnection (formerly known as CodeStar):

1. Sign in to your AFT management account
2. Locate and complete the pending AWS CodeConnections for the new VCS provider, as described in Update a pending connection [`https://us-east-1.console.aws.amazon.com/codesuite/settings/connections`]
3. Reference: Post-deployment steps

> **Note**
>
> Account pipelines retain the previous source until `aft-invoke-customizations` *Step Functions* is invoked. This invocation can be done as part of the upgrade or as part of the next customizations invocations.

For more information, see this blog: How to migrate your AWS CodeCommit repository to another Git provider.

Data protection

The AWS shared responsibility model applies to data protection in AFT. For data protection purposes, we recommend the following best practices for security.

- Follow the Data Protection guidelines provided by AWS Control Tower. For details, see Data Protection in AWS Control Tower.
- Preserve Terraform state configuration generated at the time of AFT deployment. For details, see Deploy AWS Control Tower Account Factory for Terraform (AFT).
- Rotate sensitive credentials periodically as directed by your organization's security policy. Examples of secrets are Terraform tokens, `git` tokens, and so forth.

Encryption at rest

AFT creates Amazon S3 buckets, Amazon SNS topics, Amazon SQS queues, and Amazon DynamoDB databases that are encrypted at rest with AWS Key Management Service keys. KMS keys created by AFT have yearly rotation enabled by default. If you choose the Terraform Cloud or Terraform Enterprise distributions of Terraform, AFT includes a AWS Systems Manager SecureString parameter to store Terraform token values that are sensitive.

AFT uses AWS services described in Component services that are, by default, encrypted at rest. For details, see the AWS documentation for each component AWS service of AFT, and learn about the data protection practices followed by each service.

Encryption in transit

AFT relies upon AWS services described in Component services that employ encryption in transit, by default. For details, see the AWS documentation for each component AWS service of AFT, and learn about the data protection practices followed by each service.

For Terraform Cloud or Terraform Enterprise distributions, AFT calls an HTTPS endpoint API for access to your Terraform organization. If you choose a third-party VCS provider supported by AWS CodeStar connections, AFT calls an HTTPS endpoint API for access to your VCS provider organization.

Remove an account from AFT

This topic describes how to remove an account from AFT, so the AFT pipeline stops deploying and updating the account.

> **Important**
>
> Removing an account from the AFT pipeline is irreversible and can result in a loss of state.

You might remove an account from AFT when you want to close an account for a retired application, isolate a compromised account, or move an account from one organization to another organization.

> **Note**
>
> Removing an account from AFT is different than deleting an AWS Control Tower account or AWS account. When you remove an account from AFT, AWS Control Tower still manages the account. To delete an AWS Control Tower account or AWS account, see the following:
>
> - Unmanage an account in the *AWS Control Tower User Guide*.
> - Closing an account in the *AWS Billing User Guide*.

To remove an account from the AFT pipelines

The following procedure describes how to remove an account from AFT.

1. **Remove account from `git` repository that stores account requests**

   In the `git` repository where you store account requests, delete the account request for the account you want to remove from AFT.

When you remove an account request from the account request repository, AFT deletes the customization pipeline and account metadata. For more information, see the [1.8.0 release notes] for AFT on GitHub.

2. **Delete Terraform workspace (For Terraform Cloud and Terraform Enterprise customers only)**

   Delete the global customizations and account customizations workspaces for the account that you want to remove from AFT.

3. **Delete Terraform state from Amazon S3 backend**

   In the AFT management account, delete all relevant folders inside of the Amazon S3 buckets for the account that you want to remove from AFT.

   > **Tip**
   >
   > In the following examples, replace *012345678901* with the AFT management account ID number.

   Example: Terraform OSS

   When you choose Terraform OSS, you find 3 folders for each account in the `aft-backend-012345678901-primary-region` and `aft-backend-012345678901-secondary-region` Amazon S3 buckets. These folders are related to the *account customizations state*, *customizations pipeline state*, and *global customizations state*

   Example: Terraform Cloud or Terraform Enterprise

   When you choose Terraform Cloud or Terraform Enterprise, you find a folder for each account in the `aft-backend-012345678901-primary-region` and `aft-backend-012345678901-secondary-region` Amazon S3 buckets. These folders are related to the *customizations pipeline state*.

Operational metrics

By default, *Account Factory for Terraform (AFT)* sends anonymous operational metrics to AWS. We use this data to understand how customers are using AFT so we can improve the quality and features of the solution. You can opt out of data collection by changing a parameter during AFT deployment. When collection is enabled, the following data is sent to AWS:

- **Solution:** The AFT-specific identifier
- **Version:** The version of AFT
- **Universally Unique Identifier (UUID):** Randomly generated, unique identifier for each AFT deployment
- **Timestamp:** Data-collection timestamp
- **Data:** AFT configuration and actions taken by the customer

AWS owns the data collected. Data collection is subject to the AWS Privacy Policy.

Note

Versions of AFT prior to 1.6.0 do not report usage metrics to AWS.

To opt out of reporting metrics:

- Set the input value of `aft_metrics_reporting` to `false` in your Terraform input configuration file, as shown in the example that follows, and redeploy AFT. This value is set to `true` by default, if you do not set it explicitly.

If you copy the example, remember to substitute your actual ID and Region values for the items given in strings with `x`.

```
module "control_tower_account_factory" {
source = "aws-ia/control_tower_account_factory/aws"

# Required Vars
ct_management_account_id    = "xxxxxxxxxxxx"
log_archive_account_id      = "xxxxxxxxxxxx"
audit_account_id            = "xxxxxxxxxxxx"
aft_management_account_id   = "xxxxxxxxxxxx"
ct_home_region              = "xx-xxxx-x"
tf_backend_secondary_region = "xx-xxxx-x"
```

```
    # Optional Vars
    aft_metrics_reporting = false    # to opt out, set this value to false
    }
```

Account Factory for Terraform (AFT) troubleshooting guide

This section can help you troubleshoot common issues that you might encounter when using Account Factory for Terraform (AFT).

Topics

- General issues
- Issues related to account provisioning/registration
- Issues related to customizations invocation
- Issues related to the account customizations workflow

General issues

- **Exceeded AWS resource quotas**

  If your log groups indicate that you exceeded AWS resource quotas, contact AWS Support. Account Factory uses AWS services with resource quotas that include AWS CodeBuild, AWS Organizations, and AWS Systems Manager. For more information, see the following:

  - What is AWS CodeBuild? in the *CodeBuild User Guide*.
  - What is AWS Organizations? in the *Organizations User Guide*.
  - What is AWS Systems Manager? in the *Systems Manager User Guide*.
- **Outdated version of Account Factory**

  If you encounter an issue and believe the issue is a bug, make sure that you have the latest version of Account Factory. For more information, see Updating the Account Factory version.
- **Local changes were made to the Account Factory source code**

  Account Factory is an open source project. AWS Control Tower supports the Account Factory core code. If you make a local change to the Account Factory core code, AWS Control Tower only supports your Account Factory deployment on a best-effort basis.
- **Insufficient Account Factory role permissions**

Account Factory creates IAM roles and policies to manage vended account deployments and customizations. If you change these roles or policies, the Account Factory pipeline may be unable to perform certain actions. For more information, see Required roles.

- **Account repositories not populated correctly**

  Make sure that you follow the post-deployment steps before provisioning accounts.

- **Not detecting drift after changing the OU manually**

  > **Note**
  >
  > AWS Control Tower detects drift automatically. For information about resolving drift, see Detect and resolve drift in AWS Control Tower.

  Drift isn't detected when the organizational unit (OU) is changed manually. This is due to the event-driven nature of Account Factory. When an account request is submitted, the resource that Terraform manages is an Amazon DynamoDB item, not a direct account. After an item is changed, the request is put in a queue, where AWS Control Tower processes them through Service Catalog (the service that manages account details). If you change the OU manually, drift isn't detected because the account request hasn't changed.

Issues related to account provisioning/registration

- **Account request (email address/name) already exists**

  The issue typically results in an Service Catalog product failure during provisioning or as `ConditionalCheckFailedException`.

  You can find more information about the issue by doing one of the following:

  - Review your Terraform or CloudWatch Logs log groups.
  - Review the failures that are emitted to the Amazon SNS topic `aft-failure-notifications`.

- **Malformed account request**

  Make sure that your account request follows the expected schema. For examples, see terraform-aws-control_tower_account_factory on GitHub.

- **Exceeded AWS Organizations resource quotas**

  Make sure that your account request doesn't exceed AWS Organizations resource quotas. For more information, see Quotas for AWS Organizations.

Issues related to customizations invocation

- **Target account not onboarded to Account Factory**

  Make sure all accounts that are included in a customization request have been onboarded to Account Factory. For more information, see Update an existing account.

- **Account that customization request targets exists in the DynamoDB table `aft-request-metadata`, but not in account request repository**

  Format your customization invocation request to exclude the offending account by doing one of the following:

  - In the DynamoDB table `aft-request-metadata`, delete the entry referencing the account that's no longer in your account request repository.
  - Not using "all" as the target.
  - Not targeting the OU that the account belongs to.
  - Not targeting the account directly.

- **Used incorrect token for Terraform Cloud**

  Make sure that you set up the correct token. Terraform Cloud only supports team-based tokens, not organization-based tokens.

- **Failed to create account before account customizations pipeline is created; can't customize account**

  Make a change to the account specification in the account request repository. When you make a change, such as changing a tag value for an account, Account Factory follows the path that tries to create the pipeline, even if the pipeline doesn't exist.

Issues related to the account customizations workflow

If you're experiencing issues related to the account customizations workflow, make sure that your version of AFT is 1.8.0 or higher, and that you delete all instances of account-related metadata from your DynamoDB request table.

For information about AFT version 1.8.0, see Release 1.8.0 on GitHub.

For information about how to check and update your version of AFT, see the following:

- Check the AFT version
- Update the AFT version

You can also trace and troubleshoot customization requests by using Amazon CloudWatch Logs Insights queries to filter logs containing your target account and customization request IDs. For more information, see Troubleshooting with AFT account customization request tracing.

Detect and resolve drift in AWS Control Tower

Identifying and resolving drift is a regular operations task for AWS Control Tower management account administrators. Resolving drift helps to ensure your compliance with governance requirements.

When you create your landing zone, the landing zone and all the organizational units (OUs), accounts, and resources are compliant with the governance rules enforced by your chosen controls. As you and your organization members use the landing zone, changes in this compliance status may occur. Some changes may be accidental, and some may be made intentionally to respond to time-sensitive operational events.

Drift detection assists you in identifying resources that need changes or configuration updates to resolve the drift.

Detecting drift

AWS Control Tower detects drift automatically. To detect drift, the `AWSControlTowerAdmin` role requires persistent access to your management account so AWS Control Tower can make read-only API calls to AWS Organizations. These API calls show up as AWS CloudTrail events.

Drift is surfaced in the Amazon Simple Notification Service (Amazon SNS) notifications that are aggregated in the audit account. Notifications in each member account send alerts to a local Amazon SNS topic, and to a Lambda function.

For controls that are part of the AWS Security Hub **Service-Managed Standard: AWS Control Tower**, drift is shown on the **Account** and **Account details** pages in the AWS Control Tower console, as well as by means of an Amazon SNS notification.

Member account administrators can (and as a best practice, they should) subscribe to the SNS drift notifications for specific accounts. For example, the `aws-controltower-AggregateSecurityNotifications` SNS topic provides drift notifications. The AWS Control Tower console indicates to management account administrators when drift has occurred. For more information about SNS topics for drift detection and notification, see Drift prevention and notification.

Drift notification de-duplication

If the same type of drift occurs on the same set of resources multiple times, AWS Control Tower sends an SNS notification only for the initial instance of drift. If AWS Control Tower detects that this instance of drift has been remediated, it sends another notification only if drift re-occurs for those identical resources.

Example: SCP drift is handled in the following manner

- If you modify the same managed SCP multiple times, you receive a notification for the first time you modify it.
- If you modify a managed SCP, then remediate drift, then modify it again, you'll receive two notifications.

Types of account drift

- Account moved between OUs (see *Inheritance drift on enabled baselines*)
- Account removed from organization

> **Note**
>
> When you move an account from one OU to another, the controls from the previous OU are not removed. If you enable any new hook-based control on the destination OU, the old hook-based control is removed from the account, and the new control replaces it. Controls implemented with SCPs and AWS Config rules always must be removed manually when an account changes OUs.

Examples of policy drift

- SCP updated
- SCP detached from OU

For more information, see Types of Governance Drift.

Viewing drift

You can view the drift status for your accounts and OUs through the console or APIs, and identify when account and OU configurations are *drifted*, or out of sync. Drift status also is communicated with SNS messages. For more information about receiving these SNS messages, see Guidance on subscribing to SNS Topics.

To view OU and account drift status in the console, navigate to the **Organization** page, and then select the OUs or accounts that you wish to inspect.

To view drift status for OUs and accounts programmatically, call the `ListEnabledBaselines` API to view statuses for your enabled baselines. To view statuses for individual accounts programmatically with the `ListEnabledBaselines` API, use the `includeChildren` flag. You can filter by these statuses, and see only the accounts and OUs that require your attention.

Resolving drift

Although detection is automatic, the steps to resolve drift must be done manually through the console, or with the APIs.

For example, you can resolve policy drift for controls programmatically, by calling the `ResetEnabledControl` API.

To resolve *configuration baseline drift* for an OU, you can choose **Re-register OU** in the console. If the drift is caused by a single account, you can choose **Update account** in the console. To resolve baseline drift with the APIs, you can call the `ResetEnabledBaseline` API on the OU.

Summary

- Many types of drift can be resolved through the **Landing zone settings** page. You can choose the **Reset** button in the **Versions** section to resolve these types of drift.
- If your OU has fewer than 1000 accounts, you can resolve drift in Account Factory provisioned accounts, or SCP drift, by selecting **Re-register OU** on the **Organization** page or the **OU details** page.
- You may be able to resolve account drift, such as Moved member account, by updating an individual account. For more information, see Update the account in the console.
- For controls, many types of drift can be resolved by calling the `ResetEnabledControl` API.
- Baseline drift on OUs and accounts can be resolved by calling the **ResetEnabledBaseline** API, or by choosing **Re-register OU** or **Update account** in the AWS Control Tower console.

When you take action to resolve drift on a landing zone version, two behaviors are possible.

- If you are on the latest landing zone version, when you choose **Reset** and then choose **Confirm**, your drifted landing zone resources are reset to the saved AWS Control Tower configuration. The landing zone version stays the same.
- If you are not on the latest version, you must choose **Update**. The landing zone is upgraded to the latest landing zone version. Drift is resolved as part of this process.

Considerations about drift and policys scans

AWS Control Tower scans your managed SCPs, RCPs, and Declarative policies daily to verify that the corresponding controls are applied correctly and that they have not drifted. To retrieve these resources and run checks on them, AWS Control Tower calls AWS Organizations on your behalf, using a role in your management account.

If an AWS Control Tower scan discovers drift, you'll receive a notification. AWS Control Tower sends only one notification per drift issue, so if your landing zone already is in a state of drift, you won't receive additional notifications unless a new drift item is found.

AWS Organizations limits how often each of its APIs can be called. This limit is expressed in transactions per second (TPS), and known as the *TPS limit*, *throttling rate*, or *API request rate*. When AWS Control Tower audits your SCPs, RCPs, and Dseclarative policies by calling AWS Organizations, the API calls that AWS Control Tower makes are counted towards your TPS limit, because AWS Control Tower uses the management account to make the calls.

In rare situations, this limit can be reached when you call the same APIs repeatedly, whether through a third-party solution or a custom script you wrote. For example, if you and AWS Control Tower call the same AWS Organizations APIs at the same moment in time (within 1 second), and the TPS limits are reached, subsequent calls are throttled. That is, these calls return an error such as `Rate exceeded`.

If an API request rate is exceeded

- If AWS Control Tower hits the limit and is throttled, we pause the execution of the audit and resume it at a later time.

- If your workload hits the limit and is throttled, the result can range from slight latency all the way to a fatal error in the workload, depending on how the workload is configured. This edge case is something to be aware of.

A daily SCP scan consists of

1. Retrieving your recently active OUs.
2. For each registered OU, retrieving all SCPs managed by AWS Control Tower that are attached to the OU. Managed SCPs have identifiers that begin with `aws-guardrails`.
3. For each preventive control enabled on the OU, verifying that the control's policy statement is present in the OU's managed SCPs.

An OU may have one or more managed SCPs.

Types of drift to resolve right away

Most types of drift can be resolved by administrators. A few types of drift must be resolved immediately, including deletion of an organizational unit that the AWS Control Tower landing zone requires. Here are some examples of major drift that you may wish to avoid:

- *Don't delete the Security OU:* The organizational unit originally named **Security** during landing zone setup by AWS Control Tower should not be deleted. If you delete it, you'll see an error message instructing you to reset the landing zone immediately. You won't be able to take any other actions in AWS Control Tower until the reset is complete.
- *Don't delete required roles:* AWS Control Tower checks certain AWS Identity and Access Management (IAM) roles when you log into the console for *IAM role drift*. If these roles are missing or inaccessible, you'll see an error page instructing you to reset your landing zone. These roles are `AWSControlTowerAdmin` `AWSControlTowerCloudTrailRole` `AWSControlTowerStackSetRole`.

  For more information about these roles, see [Permissions Required to use the AWS Control Tower console](#).
- *Don't delete all Additional OUs:* At least one Additional OU is required for AWS Control Tower to operate, but it doesn't have to be the **Sandbox** OU.
- *Don't remove shared accounts:* If you remove shared accounts from Foundational OUs with the AWS Organizations console or APIs, such as removing the logging account from the Security

OU, your landing zone will be in a state of drift. The landing zone must be reset before you can continue using the AWS Control Tower console. You can move an Audit or Log Archive account out of the Foundational OU, but still inside an AWS Control Tower OU, without causing drift.

Repairable changes to resources

Here's a list of changes to AWS Control Tower resources that are permitted, although they create *resolvable drift*. Results of these permitted operations are viewable in the AWS Control Tower console, although a refresh may be required.

For more information about how to resolve the resulting drift, see Managing Resources Outside of AWS Control Tower.

Changes permitted outside the AWS Control Tower console

- Change the name of a registered OU.
- Change the name of the Security OU.
- Change the name of member accounts in non-Foundational OUs.
- Change the name of AWS Control Tower shared accounts in the Security OU.
- Delete a non-Foundational OU.
- Delete an enrolled account from a non-Foundational OU.
- Change the email address of a shared account in the Security OU.
- Change the email address of a member account in a registered OU.

> **Note**
>
> Moving accounts between OUs is considered drift, and it must be resolved.

Drift and new account provisioning

If your landing zone is in a state of drift, the **Enroll account** feature in AWS Control Tower will not work. In that case, you must provision new accounts through AWS Service Catalog. For instructions, see Provision accounts with AWS Service Catalog Account Factory .

In particular, if you've made certain changes to your accounts by means of Service Catalog, such as changing the name of your portfolio, the **Enroll account** feature will not work.

Types of governance drift

Governance drift, also called *organizational drift* occurs when OUs, SCPs, and member accounts are changed or updated. The types of governance drift that can be detected in AWS Control Tower are as follows: Account and OU governance drift, Landing zone drift, Control drift for non-SCP controls, Baseline drift.

Account and OU governance drift

- Moved member account
- Removed member account
- Unplanned update to managed SCP
- SCP detached from managed OU

Landing zone drift

Another type of drift is *landing zone drift*, which may be found through the management account. Landing zone drift consists of IAM role drift, or any type of organizational drift that specifically affects Foundational OUs and shared accounts.

- Deleted Foundational OU
- Trusted access disabled

A special case of landing zone drift is *role drift*, which is detected when a required role is not available. If this type of drift occurs, the console displays a warning page and some instructions on how to restore the role. Your landing zone is unavailable until the role drift is resolved. For more information about role drift, see *Don't delete required roles* in the section called Types of drift to resolve right away.

Control drift for non-SCP controls

AWS Control Tower reports *control drift* regarding controls implemented with resource control policies (RCPs), declarative policies and controls that are part of the **AWS Security Hub Service-managed Standard: AWS Control Tower**.

- Security Hub control drift
- Control policy drift

Enabled baseline drift

When baseline configurations on member accounts are different than those applied to the parent OU, AWS Control Tower reports inheritance drift for enabled baselines (resource configurations) on your OUs and accounts. For more information about baselines, see Types of baselines.

- Inheritance drift on enabled baselines

Drift that is not reported

- AWS Control Tower does not look for drift regarding other services that work with the management account, including AWS CloudTrail, Amazon CloudWatch, IAM Identity Center, AWS CloudFormation, AWS Config, and so forth.
- AWS Control Tower does not detect resource drift or other kinds of drift that may happen if you modify the resources contained in a baseline.

Moved member account

This type of drift occurs on the account rather than the OU. This type of drift can occur when an AWS Control Tower member account, the audit account, or the log archive account is moved from a registered AWS Control Tower OU to any other OU. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that your member account 'account-
email@amazon.com (012345678909)' has been moved from organizational unit 'Sandbox
 (ou-0123-eEXAMPLE)' to 'Security (ou-3210-1EXAMPLE)'. For more information, including
 steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/
move-account'",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "ACCOUNT_MOVED_BETWEEN_OUS",
```

```
  "RemediationStep" : "Re-register this organizational unit (OU), or if the OU has more
 than 1000 accounts, you must update the provisioned product in Account Factory.",
  "AccountId" : "012345678909",
  "SourceId" : "012345678909",
  "DestinationId" : "ou-3210-1EXAMPLE"
}
```

Resolutions

When this type of drift occurs for an Account Factory provisioned account in an OU with up to 1000 accounts, you can resolve it by:

- Navigating to the **Organization** page in the AWS Control Tower console, selecting the account, and choosing **Update account** at the upper right (fastest option for individual accounts).
- Navigating to the **Organization** page in the AWS Control Tower console, then choosing **Re-register** for the OU that contains the account (fastest option for multiple accounts). For more information, see Register an existing organizational unit with AWS Control Tower.
- Updating the provisioned product in Account Factory. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.

> **Note**
>
> If you have several individual accounts to update, also see this method for making updates with a script: Provision and update accounts using automation.

- When this type of drift occurs in an OU with more than 1000 accounts, the drift resolution may depend on which type of account has been moved, as explained in the next paragraphs. For more information, see Update your landing zone.
  - **If an Account Factory provisioned account is moved –** In an OU with fewer than 1000 accounts, you can resolve the account drift by updating the provisioned product in Account Factory, by re-registering the OU, or by updating your landing zone.

    In an OU with more than 1000 accounts, you *must* resolve the drift by making an update to each moved account, either through the AWS Control Tower console or the provisioned product, because **Re-register OU** will not perform the update. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.

- **If a shared account is moved** – You can resolve the drift from moving the audit or log archive account by updating your landing zone. For more information, see Update your landing zone.

> **Deprecated field name**
>
> The field name `MasterAccountID` has been changed to `ManagementAccountID` to comply with AWS guidelines. The old name is **deprecated**. Since 2022, scripts that contain the deprecated field name no longer work.

Removed member account

This type of drift can occur when a member account is removed from a registered AWS Control Tower organizational unit. The following example shows the Amazon SNS notification when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that the member account 012345678909 has
 been removed from organization o-123EXAMPLE. For more information, including steps
 to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/remove-
account'",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "ACCOUNT_REMOVED_FROM_ORGANIZATION",
  "RemediationStep" : "Add account to Organization and update Account Factory
 provisioned product",
  "AccountId" : "012345678909"
}
```

Resolution

- When this type of drift occurs in a member account, you can resolve the drift by updating the account in the AWS Control Tower console, or in Account Factory. For example, you can add the account to another registered OU from the Account Factory update wizard. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog.

- If a shared account is removed from a Foundational OU, you must resolve the drift by resetting your landing zone. Until this drift is resolved, you will not be able to use the AWS Control Tower console.
- For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower.

Note

In Service Catalog, the Account Factory provisioned product that represents the account is not updated to remove the account. Instead, the provisioned product is displayed as `TAINTED` and in an error state. To clean up, go to the Service Catalog, choose the provisioned product, and then choose **Terminate**.

Unplanned update to managed SCP

This type of drift can occur when an SCP for a control is updated in the AWS Organizations console or programmatically using the AWS CLI or one of the AWS SDKs. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that the managed service control policy
 'aws-guardrails-012345 (p-tEXAMPLE)', attached to the registered organizational unit
 'Security (ou-0123-1EXAMPLE)', has been modified. For more information, including
 steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/
update-scp'",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "SCP_UPDATED",
  "RemediationStep" : "Update Control Tower Setup",
  "OrganizationalUnitId" : "ou-0123-1EXAMPLE",
  "PolicyId" : "p-tEXAMPLE"
}
```

Resolution

When this type of drift occurs in an OU with up to 1000 accounts, you can resolve it by:

- Navigating to the **Organization** page in the AWS Control Tower console to re-register the OU (fastest option). For more information, see Register an existing organizational unit with AWS Control Tower.
- Updating your landing zone (slower option). For more information, see Update your landing zone.

When this type of drift occurs in an OU with more than 1000 accounts, resolve it by updating your landing zone. For more information, see Update your landing zone.

SCP detached from managed OU

This type of drift can occur when an SCP for a control has been detached from an OU that's managed by AWS Control Tower. This occurrence is especially common when you're working from outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that the managed service control
 policy 'aws-guardrails-012345 (p-tEXAMPLE)' has been detached from the registered
 organizational unit 'Sandbox (ou-0123-1EXAMPLE)'. For more information, including
 steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/
scp-detached'",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "SCP_DETACHED_FROM_OU",
  "RemediationStep" : "Update Control Tower Setup",
  "OrganizationalUnitId" : "ou-0123-1EXAMPLE",
  "PolicyId" : "p-tEXAMPLE"
}
```

Resolution

When this type of drift occurs in an OU with up to 1000 accounts, you can resolve it by:

- Navigating to the OU in the AWS Control Tower console to re-register the OU (fastest option). For more information, see Register an existing organizational unit with AWS Control Tower.
- Updating your landing zone (slower option). If the drift is affecting a mandatory control, the update process creates a new service control policy (SCP) and attaches it to the OU to resolve

the drift. For more information about how to update your landing zone, see Update your landing zone.

When this type of drift occurs in an OU with more than 1000 accounts, resolve it by updating your landing zone. If the drift is affecting a mandatory control, the update process creates a new service control policy (SCP) and attaches it to the OU to resolve the drift. For more information about how to update your landing zone, see Update your landing zone.

Deleted Foundational OU

This type of drift applies only to AWS Control Tower Foundational OUs, such as the Security OU. It can occur if a Foundational OU is deleted outside of the AWS Control Tower console. Foundational OUs cannot be moved without creating this type of drift, because moving an OU is the same as deleting it and then adding it someplace else. When you resolve the drift by updating your landing zone, AWS Control Tower replaces the Foundational OU in the original location. The following example shows an Amazon SNS notification you may receive when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that the registered organizational unit
 'Security (ou-0123-1EXAMPLE)' has been deleted. For more information, including steps
 to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/delete-
ou'",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "ORGANIZATIONAL_UNIT_DELETED",
  "RemediationStep" : "Delete organizational unit in Control Tower",
  "OrganizationalUnitId" : "ou-0123-1EXAMPLE"
}
```

Resolution

Because this drift occurs for Foundational OUs only, the resolution is to update the landing zone. When other types of OUs are deleted, AWS Control Tower is updated automatically.

For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower.

Security Hub control drift

This type of drift occurs when a control that's part of the **AWS Security Hub Service-Managed Standard: AWS Control Tower** reports a state of drift. The AWS Security Hub service itself does not report a state of drift for these controls. Instead, the service sends its findings to AWS Control Tower.

Security Hub control drift also can be detected if AWS Control Tower has not received a status update from Security Hub in more than 24 hours. If those findings are not received as expected, AWS Control Tower verifies that the control is in drift. The following example shows an Amazon SNS notification you may receive when this type of drift is detected.


```
{
"Message" : "AWS Control Tower has detected that an AWS Security Hub control
 was removed in your account example-account@amazon.com <mailto:example-
account@amazon.com>. The artifact deployed on the target OU and accounts does not match
 the expected template and configuration for the control. This mismatch indicates that
 configuration changes were made outside of AWS Control Tower. For more information,
 view Security Hub standard",
"MasterAccountId" : "123456789XXX",
"ManagementAccountId" : "123456789XXX",
"OrganizationId" : "o-123EXAMPLE",
"DriftType" : "SECURITY_HUB_CONTROL_DISABLED",
"RemediationStep" : "To remediate the issue, Re-register the OU, or remove the control
 and enable it again. If the problem persists, contact AWS support.",
"AccountId" : "7876543219XXX",
"ControlId" : "SH.XXXXXXX.1",
"ControlName" : "EBS snapshots should not be publicly restorable",
"ApiControlIdentifier" : "arn:aws:controltower:us-east-1::control/PYBETSAGNUZB",
"EnabledControlIdentifier": "arn:aws:controltower:us-
east-1::enabledcontrol/<UNIQUE_ID>".
"Region" : "us-east-1"
}
```


Resolution

For OUs with fewer than 1000 accounts, the recommended resolution is to call the **ResetEnabledControl** API for the drifted control. In the console, you can select **Re-register** for the OU, which resets the control to the original state. Alternatively, for any OU, you can remove and re-enable the control through the console or the AWS Control Tower APIs, which also resets the control.

For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower.

Control policy drift

This type of drift occurs when a control that's implemented with *resource control policies* (RCPs) or *declarative policies* reports a state of drift. It returns a state of `CONTROL_INEFFECTIVE`, which you can view in the AWS Control Tower console and in the drift message. The drift message for this type of drift also includes the `EnabledControlIdentifier` for the affected control.

This type of drift is not reported for SCP-based controls.

The following example shows an Amazon SNS notification you may receive when this type of drift is detected.

```
{
    "Message": "AWS Control Tower detects that a policy it owns was updated
 unexpectedly. This mismatch indicates that configuration changes were made outside of
 AWS Control Tower.",
    "MasterAccountId": "123456789XXX",
    "ManagementAccountId": "123456789XXX",
    "OrganizationId": "o-123EXAMPLE",
    "DriftType": "CONTROL_INEFFECTIVE",
    "RemediationStep": "To remediate the issue, Reset the DRIFTED enabled control if
 permitted or Re-register the OU. If the problem persists, contact AWS support.",
    "TargetIdentifier": "arn:aws:::organizations/o-123456/ou-1234-4567",
    "ControlId": "CT.XXXXXXX.PV.1",
    "ControlName": "EBS snapshots should not be publicly restorable",
    "ApiControlIdentifier": "arn:aws:controlcatalog:::control/<UNIQUE_ID>",
    "EnabledControlIdentifier": "arn:aws:controltower:us-
east-1::enabledcontrol/<UNIQUE_ID>"
}
```

Resolution

The easiest resolution for control policy drift on RCP controls, declarative policy controls, and Security Hub controls enabled in AWS Control Tower is to call the `ResetEnabledControl` API.

For OUs with fewer than 1000 accounts, another resolution from the console or API is to **Re-register** the OU, which resets the control to the original state.

For any individual OU, you can remove and re-enable the control through the console or the AWS Control Tower APIs, which also resets the control.

For more information about resolving drift for accounts and OUs, see [If you manage resources outside of AWS Control Tower](#).

Trusted access disabled

This type of drift applies to AWS Control Tower landing zones. It occurs when you disable trusted access to AWS Control Tower in AWS Organizations after you set up your AWS Control Tower landing zone.

When trusted access is disabled, AWS Control Tower no longer receives change events from AWS Organizations. AWS Control Tower relies on these change events to stay synchronized with AWS Organizations. As a result, AWS Control Tower may miss organizational changes in accounts and OUs. That is why it is important to re-register each OU, each time you update your landing zone.

Example: Amazon SNS notification

The following is an example of the Amazon SNS notification that you receive when this type of drift occurs.

```
{
  "Message" : "AWS Control Tower has detected that trusted access has been disabled in
 AWS Organizations. For more information, including steps to resolve this issue, see
 https://docs.aws.amazon.com/controltower/latest/userguide/drift.html#drift-trusted-
access-disabled",
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "TRUSTED_ACCESS_DISABLED",
  "RemediationStep" : "Reset Control Tower landing zone."
}
```

Resolution

AWS Control Tower notifies you when this type of drift occurs in the AWS Control Tower console. The resolution is to reset your AWS Control Tower landing zone. For more information, see [Resolving drift](#).

Inheritance drift on enabled baselines

This type of drift can occur to AWS Control Tower OUs and accounts.

Resolution

AWS Control Tower notifies you when this type of drift occurs. For almost all cases of inheritance drift, you will receive an SNS notification for *Moved member account* drift. That's because this type of drift typically occurs when an account has been moved, or an account fails enrollment.

View and resolve drift in the console

In the AWS Control Tower console, you can view the drift status in the **Baseline state** column on the **Organizations** page. The resolution from the console is to **Re-register** your OU or **Update** your account.

View and resolve drift programmatically

To view drift status programmatically, you can call the `ListEnabledBaselines` API to view statuses for the enabled baselines on your OUs. To view statuses for individual accounts programmatically with the `ListEnabledBaselines` API, use the `includeChildren` flag.

You can resolve this type of drift programmatically, by calling the `ResetEnabledBaseline` API.

If you manage resources outside of AWS Control Tower

AWS Control Tower sets up accounts, organizational units, and other resources on your behalf, but you are the owner of these resources. You can change these resources within AWS Control Tower or outside it. The most common place to change resources outside of AWS Control Tower is the AWS Organizations console. This topic describes how to reconcile changes to AWS Control Tower resources when you make the changes outside of AWS Control Tower.

Renaming, deleting, and moving resources outside of the AWS Control Tower console causes the console to become out of sync. Many changes can be reconciled automatically. Certain changes require a reset to your landing zone, to update the information that's displayed in the AWS Control Tower console.

In general, changes that you make outside the AWS Control Tower console to AWS Control Tower resources create a state of *resolvable drift* in your landing zone. For more information about these changes, see Repairable changes to resources.

Tasks that require landing zone reset

- Deleting the Security OU *(A special case, not to be done lightly.)*

- Removing a shared account from the Security OU *(Not recommended.)*
- Updating, attaching, or detaching an SCP associated with the Security OU.

Changes that are updated automatically by AWS Control Tower

- Changing the email address of an enrolled account
- Renaming an enrolled account
- Creating a new top-level organizational unit (OU)
- Renaming a registered OU
- Deleting a registered OU *(Except the Security OU, which requires an update.)*
- Deleting an enrolled account *(Except a shared account in the Security OU.)*

> **Note**
>
> AWS Service Catalog handles changes differently than AWS Control Tower. AWS Service Catalog may create a change in governance posture when it reconciles your changes. For more information about updating a provisioned product, see Updating Provisioned Products in the AWS Service Catalog documentation.

Referring to resources outside of AWS Control Tower

When you create new OUs and accounts outside of AWS Control Tower, they are not governed by AWS Control Tower, even though they may be displayed.

Creating an OU

Organizational Units (OUs) created outside of AWS Control Tower are referred to as *Unregistered*. They are displayed in the **Organization** page, but they are not governed by AWS Control Tower controls.

Creating an account

Accounts created outside of AWS Control Tower are referred to as *Unenrolled*. Enrolled and unenrolled accounts that belong to an OU that's registered with AWS Control Tower are displayed in the **Organization** page. Accounts that do not belong to a registered OU can be invited by using the AWS Organizations console. This invitation to join does not enroll the account in AWS Control

Tower or extend AWS Control Tower governance to the account. To extend governance by enrolling the account, go to the **Organization** page or the **Account detail** page in AWS Control Tower and choose **Enroll account**.

Externally changing AWS Control Tower resource names

You can change the names of your organizational units (OUs) and accounts outside of the AWS Control Tower console, and the console updates automatically to reflect those changes.

Renaming an OU

In AWS Organizations, you can change the name of an OU by using either the AWS Organizations API or the console. When you change an OU name outside of AWS Control Tower, the AWS Control Tower console automatically reflects the name change. However, if you provision your accounts using AWS Service Catalog, you also must reset your landing zone to ensure that AWS Control Tower stays consistent with AWS Organizations. The **Reset** workflow ensures consistency across services for the Foundational and Additional OUs. You can resolve this type of drift from the **Landing zone settings** page. See the section called "Resolving Drift" in Detect and resolve drift in AWS Control Tower.

AWS Control Tower displays the names of OUs on the **Organization** page in the AWS Control Tower dashboard. You can see when your landing zone reset operation has succeeded.

Renaming an enrolled account

Each AWS account has a display name that can be changed by the account's root user in the AWS Billing and Cost Management console. When you rename an account that's enrolled in AWS Control Tower, the name change is automatically reflected in AWS Control Tower. For more information about changing an account's name, see Managing an AWS account in the *AWS Billing User Guide*.

Deleting the Security OU

This type of drift is a special case. If you delete the **Security** OU, you will see an error message page, prompting you to reset your landing zone. You must reset your landing zone before you can take any other actions in AWS Control Tower.

- You will not be able to perform any actions in the AWS Control Tower console and you will not be able to create any new accounts in AWS Service Catalog until the reset is done.
- You won't be able to view the **Landing zone settings** page to see the **Reset** button there.

In this situation, the landing zone reset process creates a new Security OU and moves the two shared accounts into the new Security OU. AWS Control Tower marks the Log Archive and Audit accounts as drifted. The same process resolves the drift in these accounts.

If you determine that you must delete the Security OU, here's what you need to know:

Before you can delete the **Security** OU, you must make sure it contains no accounts. Specifically, you must remove the Log Archive and Audit accounts from the OU. We recommend that you move these accounts to another OU.

> **Note**
>
> The action of deleting your Security OU is not to be performed without due consideration. The action could create compliance concerns if logging is suspended temporarily, and because some controls might not be enforced.

For general information about drift, see "Resolving Drift" in [Detect and resolve drift in AWS Control Tower](#).

Removing an account from the Security OU

We do not recommend that you remove any of the shared accounts from your organization or move them out of the **Security** OU. If you have removed a shared account accidentally, you can follow the remediation steps in this section to restore the account.

- **From within the AWS Control Tower console:** To start the remediation process, follow the semi-manual remediation steps. Ensure the user or role you use to access the AWS Control Tower console has permissions to run `organizations:InviteAccountToOrganization`. If you don't have such permissions, follow the manual remediation steps, which use both the AWS Control Tower console and the AWS Organizations console.
- **Starting from the AWS Organizations console:** This remediation process is a slightly longer, fully manual procedure. When following the manual remediation steps, you'll switch between the AWS Organizations console and the AWS Control Tower console. When working in AWS Organizations, you'll need a user or role with the `AWSOrganizationsFullAccess` managed policy or equivalent. When working in the AWS Control Tower console, you'll need a user or role with the `AWSControlTowerServiceRolePolicy` managed policy or equivalent, and permission to run all AWS Control Tower actions (controltower:*).

- If the remediation steps don't restore the account, contact AWS Support.

The results of removing a shared account through AWS Organizations:

- The account is no longer protected by AWS Control Tower mandatory controls with service control policies (SCPs). **Result:** *The resources created by AWS Control Tower in the account may be modified or deleted.*
- The account is no longer under the AWS Organizations management account. **Result:** *The administrator of the AWS Organizations management account no longer has visibility into the account's spending.*
- The account is no longer guaranteed to be monitored by AWS Config. **Result:** *The administrator of the AWS Organizations management account may not be able to detect resource changes.*
- The account is no longer in the organization. **Result:** *AWS Control Tower updates and reset will fail.*

To restore a shared account using the AWS Control Tower console (semi-manual procedure)

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower. You must sign in as an IAM user, user in IAM Identity Center, or role with permissions to run `organizations:InviteAccountToOrganization`. If you don't have such permissions, use the manual remediation procedure described later in this topic.
2. On the **Landing zone drift detected** page, choose **Re-Invite** to remediate shared account removal by re-inviting the shared account into the organization. An automatically-generated email is sent to the email address for the account.
3. Accept the invitation to bring the shared account back into the organization. Do one of the following:

   - Sign in to the shared account that was removed, then go to https://console.aws.amazon.com/organizations/home#/invites
   - If you have access to the email message sent when you re-invited the account, sign in to the removed account, then click the link in the message to navigate directly to the account invitation.
   - If the shared account that was removed is not in another organization, sign into the account, open the AWS Organizations console and navigate to **Invitations**.

4. Sign in to the management account again, or reload the AWS Control Tower console if it's already open. You'll see the **Landing zone drift** page. Choose **Reset** to repair the landing zone.
5. Wait for the reset process to complete.

If remediation is successful, the shared account appears in a normal state and compliance.

If the remediation steps don't restore the account, contact AWS Support.

To restore a shared account using the AWS Control Tower and AWS Organizations consoles (Manual remediation)

1. Sign in to the AWS Organizations console at https://console.aws.amazon.com/organizations/. You must sign in as an IAM user, user in IAM Identity Center, or role with the `AWSOrganizationsFullAccess` managed policy or equivalent.
2. Invite the shared account back to the organization. For information on the requirements, prerequisites, and procedure for inviting an account to AWS Organizations, see Inviting an AWS account to your organization in the *AWS Organizations User Guide*.
3. Sign in to the shared account that was removed, then go to https://console.aws.amazon.com/organizations/home#/invites to accept the invitation.
4. Sign in to the management account again.
5. Sign in to the AWS Control Tower console as a user or role with the `AWSControlTowerServiceRolePolicy` managed policy or equivalent, and permissions to run all AWS Control Tower actions (controltower:*).
6. You'll see the **Landing zone drift** page with an option to reset the landing zone. Choose **Reset** to repair the landing zone.
7. Wait for the reset process to complete.

If remediation is successful, the shared account appears in a normal state and compliance.

If the remediation steps don't restore the account, contact AWS Support.

External changes that are updated automatically

Changes that you make to your account email addresses are updated by AWS Control Tower automatically, but Account Factory does not update them automatically.

Changing the email address of a governed account

AWS Control Tower retrieves and displays email addresses as required by the console experience. Therefore, shared and other account email addresses are updated and shown consistently in AWS Control Tower after you change them.

> **Note**
>
> In AWS Service Catalog, the Account Factory displays the parameters that were specified in the console when you created a provisioned product. However, the original account email address is not updated automatically when the account email address changes. That's because the account is conceptually contained within the provisioned product; it is not the same as the provisioned product. To update this value, you must update the provisioned product, which may cause a change in governance posture.

Applying external AWS Config rules

AWS Control Tower displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower, including rules that were activated outside of the AWS Control Tower console.

Deleting AWS Control Tower resources outside AWS Control Tower

You can delete OUs and accounts in AWS Control Tower and you don't need to take any further action to see the updates. Account Factory is updated automatically when you delete an OU, but not when you delete an account.

Deleting a registered OU (except the Security OU)

Within AWS Organizations, you can remove empty organizational units (OUs) by using the API or the console. OUs that contain accounts cannot be deleted.

AWS Control Tower receives a notification from AWS Organizations when an OU is deleted. It updates the OU list in the Account Factory, so that the list of registered OUs remains consistent.

> **Note**
>
> In AWS Service Catalog, the Account Factory is updated to remove the deleted OU from the list of available OUs into which you can provision an account.

Deleting an enrolled account from an OU

When you delete an enrolled account, AWS Control Tower receives a notification and makes updates, so that the information remains consistent.

Note

In AWS Service Catalog, the Account Factory provisioned product that represents the governed account is not updated to delete the account. Instead, the provisioned product is displayed as `TAINTED` and in an error state. To clean up, go to AWS Service Catalog, choose the provisioned product, and then choose **Terminate**.

Govern organizations and accounts with AWS Control Tower

All organizational units (OUs) and accounts that you create in AWS Control Tower are governed automatically by AWS Control Tower. Also, if you have existing OUs and accounts that were created outside of AWS Control Tower, you can bring them into AWS Control Tower governance.

For existing AWS Organizations and AWS accounts, most customers prefer to enroll groups of accounts by registering the entire organizational unit (OU) that contains the accounts. You also can enroll accounts individually. For more information on enrolling individual accounts, see Enroll an existing AWS account.

Terminology

- When you bring an existing organization into AWS Control Tower, it's called *registering* the organization, or *extending governance* to the organization.
- When you bring an AWS account into AWS Control Tower, it's called *enrolling* the account.

View your OUs and accounts

On the AWS Control Tower **Organization** page, you can view all the OUs in your AWS Organizations, including OUs that are registered with AWS Control Tower and those that are not registered. You can view nested OUs as part of the hierarchy. An easy way to view your organizational units on the **Organization** page is to select **Organizational units only** from the dropdown at the upper right.

The **Organization** page lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. An easy way to view your accounts on the **Organization** page is to select **Accounts only** from the dropdown at the upper right. You can view, update, and enroll accounts individually within the OUs, if the accounts meet the prerequisites for enrollment.

If you do not select any filtering, the **Organization** page displays your accounts and OUs in a hierarchy. It is a central location for monitoring and taking actions on all of your AWS Control Tower resources. For more information about the **Organization** page, you can view the video walkthrough.

Video walkthrough

This video (4:01) describes how to work with the **Organization** page in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Working with the Organization Page in AWS Control Tower.

Topics

Extend governance to an existing organization

You can add AWS Control Tower governance to an existing organization by setting up a landing zone (LZ) as outlined in the AWS Control Tower User Guide at Getting Started, Step 2.

Here's what to expect when you set up your AWS Control Tower landing zone in an existing organization.

- You can have one landing zone per AWS Organizations organization.
- AWS Control Tower uses the management account from your existing AWS Organizations organization as its management account. No new management account is needed.
- AWS Control Tower sets up two new accounts in a registered OU: an audit account and a logging account.
- Your organization's service limits must allow for the creation of these two additional accounts.
- After you've launched your landing zone or registered an OU, AWS Control Tower controls apply automatically to all enrolled accounts in that OU.
- You can **Enroll** additional existing AWS accounts into an OU that's governed by AWS Control Tower, so that controls apply to those accounts.
- You can add more OUs in AWS Control Tower and you can **Register** existing OUs.

To check other prerequisites for registration and enrollment, see Getting Started with AWS Control Tower.

Here's more detail about how AWS Control Tower controls **do not** apply to your OUs in AWS organizations that don't have AWS Control Tower landing zones set up:

- New accounts created outside of AWS Control Tower Account Factory are not bound by the registered OU's controls.
- New accounts created in OUs that are not registered with AWS Control Tower are not bound by controls, unless you specifically **Enroll** those accounts into AWS Control Tower. See Enroll an existing AWS account for more information about enrolling accounts.
- Additional existing organizations, existing accounts, and any new OUs or any accounts that you create outside of AWS Control Tower, are not bound by AWS Control Tower controls, unless you separately register the OU or enroll the account.

For more information about how to apply AWS Control Tower to existing OUs and accounts, see Register an existing organizational unit with AWS Control Tower.

For an overview of the process of setting up an AWS Control Tower landing zone in your existing organization, see the video in the next section.

> **Note**
>
> During set up, AWS Control Tower performs pre-checks to avoid common issues. However, if you are currently using the AWS Landing Zone solution for AWS Organizations, check with your AWS solutions architect before you try to enable AWS Control Tower in your organization to determine if AWS Control Tower may interfere with your current landing zone deployment. Also, see If the account does not meet the prerequisites for information about moving accounts from one landing zone to another.

Video: Enable a Landing Zone in existing AWS Organizations

This video (7:48), describes how to set up and enable an AWS Control Tower landing zone in existing AWS Organizations structures. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Enable AWS Control Tower for existing organizations

Considerations for IAM Identity Center and existing organizations

- If AWS IAM Identity Center (IAM Identity Center) is already set up, the AWS Control Tower home Region must be the same as the IAM Identity Center Region.
- AWS Control Tower does not delete an existing configuration.
- If IAM Identity Center is already enabled, and if you are using IAM Identity Center Directory, AWS Control Tower adds resources such as permission sets, groups, and so forth, and proceeds as usual.
- If another directory (external, AD, Managed AD) is set up, AWS Control Tower does not change the existing configuration. For more details, see Considerations for AWS IAM Identity Center (IAM Identity Center) customers.

Access to other AWS services

After you bring your organization into AWS Control Tower governance, you still have access to any AWS services that are available through AWS Organizations, by means of the AWS Organizations console and APIs. For more information, see Related AWS services.

Nested OUs in AWS Control Tower

This chapter lists the expectations and considerations you'll want to be aware of when working with nested OUs in AWS Control Tower. In most ways, working with nested OUs is the same as working with a flat OU structure. The **Register** and **Re-register** features work with nested OUs, except for the changed behaviors that are noted in this chapter.

Video Walkthrough

This video (4:46) describes how to manage nested OU deployments in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Managing Nested OUs in AWS Control Tower.

For guidance regarding best practices for nested OUs and your landing zone, see the blog post Organizing your AWS Control Tower landing zone with nested OUs.

Expand from flat OU structure to nested OU structure

If you created your AWS Control Tower landing zone with a flat OU structure, you can expand it to a nested OU structure.

This process has four main steps:

1. Create your desired nested OU structure in AWS Control Tower.
2. Go to the AWS Organizations console and use their bulk move feature to move the accounts from the source OU (flat) into the destination OU (nested). Here's how:
   a. Go to the OU from which you want to move accounts.
   b. Select all the accounts in the OU.
   c. Choose **Move**.

   > **Note**
   >
   > This step must be done in the in AWS Organizations console because AWS Control Tower doesn't have a **Move** feature.

3. Go to the nested OU in AWS Control Tower and **Register** or **Re-register** it. All of the accounts in the nested OU will be enrolled.
   - If you created the OU in AWS Control Tower, **Re-register** the OU.
   - If you created the OU in AWS Organizations, **Register** the OU for the first time.
4. After your accounts are moved and enrolled, delete the empty top-level OU, either from the AWS Organizations console or from the AWS Control Tower console.

Nested OU registration pre-checks

To support successful registration of your nested OUs and their member accounts, AWS Control Tower performs a series of pre-checks. These same prechecks are performed when registering any top-level OU or nested OU. For more information, see Common causes of failure during registration or re-registration.

- If all pre-checks pass, AWS Control Tower begins registering your OU, automatically.
- If any pre-checks fail, AWS Control Tower stops the registration process and provides you with a list of items that must be fixed before you can register your OU.

Nested OUs and roles

AWS Control Tower deploys the `AWSControlTowerExecution` role to accounts under the target OU, and to accounts in all OUs nested under the target OU, even when your intention is to register the target OU only. This role gives any user of the management account **Administrator** permissions on any account that has the `AWSControlTowerExecution` role. The role can be used to perform actions that normally would be disallowed by AWS Control Tower controls.

You can delete this role from unenrolled accounts that you don't plan to enroll. If you delete this role, you cannot enroll the account with AWS Control Tower, or register the immediate parent OUs, unless you restore the role to the account. To delete the `AWSControlTowerExecution` role from an account, you must be signed in under the `AWSControlTowerExecution` role, because no other IAM principals are allowed to delete roles managed by AWS Control Tower.

For information about how to restrict role access, see Optional conditions for your role trust relationships.

What happens during registration and re-registration of nested OUs and accounts

When you register or re-register a nested OU, AWS Control Tower enrolls all unenrolled accounts of the target OU, and it updates all enrolled accounts. Here's what to expect.

AWS Control Tower performs the following tasks

- Adds the `AWSControlTowerExecution` role to all unenrolled accounts under this OU, and to all unenrolled accounts in its nested OUs.
- Enrolls member accounts that are not enrolled.
- Re-enrolls enrolled member accounts.
- Creates an IAM Identity Center login for newly enrolled member accounts.
- Updates existing enrolled member accounts to reflect your landing zone changes.
- Updates controls that are configured for this OU and its member accounts.

Considerations for nested OU registration

- You cannot register an OU under the core OU (Security OU).
- Nested OUs must be registered separately.

- You cannot register an OU unless its parent OU is registered.
- You cannot register an OU unless all OUs higher in the tree have been registered successfully at some time (some may have been deleted).
- You can register an OU that is under a drifted higher OU, but the drift is not repaired by that action.

Nested OU limitations

- OUs may be nested a maximum of 5 levels deep under the root.
- Nested OUs under the target OU must be registered or re-registered separately.
- If the target OU is at Level 2 or below in the hierarchy, that is, if it is not a top-level OU, preventive controls enabled on higher OUs are enforced on this OU and all OUs below it, automatically.
- OU registration failures do not propagate up the hierarchy tree. You can see details about the states of nested OUs on the parent's OU details page.
- OU registration failures do not propagate down the hierarchy tree.
- AWS Control Tower does not modify your VPC settings for any new or existing accounts.

Nested OUs and compliance

From the AWS Control Tower console, you can view OUs and accounts that are non-compliant in the **Organization** page, so you can understand compliance at a larger scale.

Considerations about compliance for nested OUs and accounts

- An OU's compliance is not determined based on the compliance of the OUs nested under it.
- A control's compliance status is computed over all OUs on which the control is enabled, including nested OUs. See AWS Control Tower compliance status for OUs and accounts.
- An OU is shown as noncompliant only if it has accounts that are noncompliant, regardless of where the OU sits in the OU hierarchy.
- If a nested OU is noncompliant, its parent OU is not automatically considered to be noncompliant.
- On the **OU detail** or **Account detail** page, you can view a list of noncompliant resources that may be causing your OUs or accounts to show a non-compliant status.

Nested OUs and drift

In certain situations, drift can prevent the registration of nested OUs.

Expectations for drift and nested OUs

- You can enable controls on OUs with drifted parents, but not on drifted OUs directly.
- You are allowed to enable detective controls under a drifted OU, as long as it's not a top-level drifted OU.
- Mandatory controls are enabled on top-level OUs only. Mandatory controls are skipped when you register a nested OU.
- One mandatory control protects AWS Config resources; therefore, that control must be in a non-drifted state to register nested OUs. If drifted, AWS Control Tower blocks registration of nested OUs.
- If the top-level OU is in drift, the control that protects AWS Config resources may be in drift. In this situation, AWS Control Tower blocks any action that requires creation or update of AWS Config resources, including application of detective controls.

Nested OUs and controls

When you enable a control on a registered OU, preventive and detective controls have different behaviors. For nested OUs, proactive controls behave similarly to detective controls.

Preventive controls

- Preventive controls are enforced on nested OUs.
- Mandatory preventive controls are enforced on all accounts under the OU and its nested OUs.
- Preventive controls affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered.

Detective and proactive controls

- Nested OUs do not inherit detective or proactive controls automatically; these must be enabled separately.
- Detective and proactive controls are deployed only to registered accounts in your landing zone's operating Regions.

Enabled control states and inheritance

You can view inherited controls for each OU, on the **OU details** page.

> **Tip**
>
> You can make use of control inheritance to help stay within an OU's SCP quota. For example, you can enable a control at the top-level OU of an OU hierarchy, instead of enabling directly for a nested OU.

Inherited status

- The status **Inherited** indicates that the control is enabled by inheritance only, and it has not been applied directly to the OU.
- The status **Enabled** means the control is enforced on this OU, regardless of its state on other OUs.
- The status **Failed** means the control is not enforced on this OU, regardless of its state on other OUs.

> **Note**
>
> The status **Inherited** indicates that the control was applied to an OU higher in the tree, and it is enforced on this OU, but it was not added directly to this OU.

> **If your landing zone is not the current version**
>
> Each row in the **Enabled controls** table represents one enabled control on one, individual OU.

Nested OUs and the root

The root is not an OU, and it cannot be registered or re-registered. You also can’t create accounts directly in the root. The root cannot be noncompliant or have a lifecycle state, such as *registered* or *in drift*.

However, the root is the top-level container for all accounts and OUs. In the context of nested OUs, it is the node under which all other OUs are nested.

Register an existing organizational unit with AWS Control Tower

An efficient way to bring multiple, existing AWS accounts into AWS Control Tower is to *extend governance* by AWS Control Tower to an entire organizational unit (OU).

To enable AWS Control Tower governance over an existing OU that was created with AWS Organizations, and its accounts, *register* the OU with your AWS Control Tower landing zone. You can register OUs that contain up to 1000 accounts. If an OU contains more than 1000 accounts, you cannot register it in AWS Control Tower.

When you register an OU, its member accounts are enrolled into the AWS Control Tower landing zone. They are governed by the controls that apply to their OU.

> **Note**
>
> If you don't already have an AWS Control Tower landing zone, start by setting up a landing zone, either in a new organization created by AWS Control Tower, or in an existing AWS Organizations organization. For more details about how to set up a landing zone, see [Getting started with AWS Control Tower](#).

What happens to my accounts when I register my OU?

AWS Control Tower requires permission to establish trusted access between AWS CloudFormation and AWS Organizations on your behalf, so that AWS CloudFormation can deploy your stack to the accounts in your organization automatically.

- The `AWSControlTowerExecution` role is added to all accounts with status **Not enrolled**.
- Mandatory controls are enabled by default to your OU and all its accounts when you register your OU.

Partial enrollment of accounts after an OU is registered

It's possible to register an OU successfully, yet certain accounts may remain unenrolled. If so, these accounts do not meet some of the prerequisites for enrollment. If an account enrollment as part of the **Register OU** process does not succeed, the account status on the accounts page shows **Enrollment failed**. You may also see account information on your OU page such as **4 of 5**, in the accounts field.

For example, if you see **4 of 5**, it means that your OU has 5 accounts in total, and 4 of them enrolled successfully, but one account failed to enroll during the **Register OU** process. You can choose **Re-Register OU** to bring accounts into enrollment, after you make sure the accounts meet the enrollment prerequisites.

IAM user prerequisites for registering an OU

Your AWS Identity and Access Management (IAM) identity (user or role) or IAM Identity Center user identity must be included on the appropriate Account Factory portfolio when you perform the **Register OU** operation, even if you already have `Admin` permissions. Otherwise, the creation of the provisioned products will fail during registration. Failure occurs because AWS Control Tower relies upon the credentials of the IAM user or IAM Identity Center user identity when registering an OU.

The relevant portfolio is one created by AWS Control Tower, called **AWS Control Tower Account Factory Portfolio**. Navigate to it by choosing **Service Catalog > Account Factory > AWS Control Tower Account Factory Portfolio**. Then select the tab called **Groups, roles, and users** to view your IAM or IAM Identity Center identity. For more information on how to grant access, see [the documentation for AWS Service Catalog.]()

Register an existing OU

In the AWS Control Tower console, on the **Organization** page, you can view all of of your organization's OUs and accounts in a hierarchy, including OUs that are registered with AWS Control Tower, and those that are not registered.

In general, unregistered OUs were created in AWS Organizations, and they are not governed by any other landing zone. You can register existing OUs that contain up to 1000 accounts. If an OU contains more than 1000 accounts, you cannot register it in AWS Control Tower.

To register an existing OU from the console

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. In the left-pane navigation menu, choose **Organization**.

3. On the **Organization** page, select the radio button next to the OU you want to register, then select **Register organizational unit** from the **Actions** dropdown menu at the upper right, or alternatively, select the name of the OU so you can view the **OU details** page for that OU.

4. On the **OU details** page, at the upper right you can select **Register OU** from the **Actions** dropdown menu.

The registration process takes a minimum of 10 minutes to extend governance to the OU, and up to 2 additional minutes for each additional account.

To register an existing OU with APIs

To register an existing OU with the AWS Control Tower APIs, you can call the `EnableBaseline` API with the `AWSControlTowerBaseline` in the `baselineIdentifier` field. For more information, see Register an AWS Control Tower OU with APIs only.

Results of registering an existing OU

After you register an existing OU, the `AWSControlTowerExecution` role allows AWS Control Tower to extend governance to its individual accounts. Guardrails are enforced, and information about account activities is reported to your audit and logging accounts.

Other results include the following:

- `AWSControlTowerExecution` allows auditing by the AWS Control Tower audit account.
- `AWSControlTowerExecution` helps you configure your organization's logging, so that all the logs for every account are sent to the logging account.
- `AWSControlTowerExecution` ensures that your selected AWS Control Tower controls apply automatically to every individual account in your OUs, as well as to every new account you create in AWS Control Tower.

For a registered OU, you can provide compliance and security reports based on the auditing and logging features embodied by AWS Control Tower controls. Your security and compliance teams can verify that all requirements are met, and that no organizational drift has occurred. For more information about drift, see Detect and resolve drift in AWS Control Tower.

> **Note**
>
> One unusual situation can occur when AWS Control Tower displays OUs and their accounts. If you have created an account in a registered OU and then you subsequently move that enrolled account into another OU that's not registered, particularly if you use AWS Organizations to move the account, you can see a result "1 of 0" accounts in your OU details page. Furthermore, you may have created another unenrolled account in that unregistered OU. If there's an unregistered account, the console may read "1 of 1" for the OU. It will seem that the single (newly created) account is enrolled, but in fact it is not. You must enroll the new account.

Create a new OU

Here's how to create an OU or a nested OU in AWS Control Tower.

To create a new OU in AWS Control Tower

1. Navigate to the **Organization** page.
2. Select **Create organizational unit** from the **Create resources** dropdown menu in the upper right.
3. Specify a name in the **OU name** field.
4. In the **Parent OU** dropdown, you can see the hierarchy of registered OUs. Select a parent OU for the new OU you're creating.
5. Choose **Add**.

> **Tip**
>
> To add a nested OU in fewer steps, select the name of the parent OU shown in the table on the **Organization** page, view the **OU** page for that parent OU, and then choose **Add an OU** from the **Actions** dropdown menu in the upper right. The new OU is created as a nested OU under your selected OU, automatically.

> **Note**
>
> If your landing zone is not up to date, you will see a flat list instead of a hierarchy in the dropdown menu. Even if your landing zone includes nested OUs, you will not see L5 OU's in the dropdown, because you cannot create a new OU beneath a L5 OU. For more information about nested OUs in AWS Control Tower, see Nested OUs in AWS Control Tower.

Remove an OU

AWS Control Tower supports separate console actions to *deregister* an OU and to *delete* an OU.

Considerations

- The OU must be empty of accounts for **Delete** and **Deregister** operations to succeed.
- All optional controls must be removed from the OU.
- You must deregister the OU before you delete it.
- You can remove an OU from AWS Control Tower by deregistering it, without deleting it.

To remove an OU from AWS Control Tower

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. Navigate to the **Organization** page.
3. Select the name of the OU to view the **OU details** page, and be sure that all accounts are removed from the OU.
4. Also on the **OU details** page, be sure that all optional controls are removed from the OU.
5. Return to the **Organization** page and select the radio button next to the OU.
6. Select **Deregister organizational unit** from the **Actions** dropdown menu in the upper right.
7. **Stop here** if you do not wish to delete the OU entirely, only to deregister it from AWS Control Tower. To delete the OU completely, continue to the next step.
8. To continue, select **Delete** from the **Actions** dropdown menu in the upper right.

You must wait until the deregistration process is complete before you can deregister another OU.

> **Note**
>
> To remove accounts managed by AWS Control Tower, you can navigate to **Account factory** from the left navigation pane in the AWS Control Tower console. To remove accounts in the OU that are not managed by AWS Control Tower, go to the AWS Organizations console.

To deregister an OU programmatically, call the `DisableBaseline` API.

Common causes of failure during registration or re-registration

In general, when you register or re-register an OU, all accounts within that OU are enrolled in AWS Control Tower. However, it is possible that some accounts may fail to enroll, even if the OU as a whole is registered successfully. In these cases, you must resolve the pre-check failure related to the account and then try re-enrolling that account or OU.

If registration (or re-registration) of an OU or any of its member accounts fails, AWS Control Tower returns error messages for the member accounts that are affected. You can view the error messages on the **OU details** page, where a table aggregates the prechecks and account error messages. If a **Register OU** operation fails, the table shows all of the error messages for all of the accounts under the OU. If needed, you also can view the error messages on the **Account details** page for each account.

Optionally, you can download a file containing a detailed report that shows which pre-checks did not pass, for offline analysis. You can complete the download by choosing the **Download** button, which appears at the upper right of the registration area.

This section lists the types of errors you may receive if pre-checks fail, and how to correct the errors.

Landing Zone error

- **Landing zone not ready**

  Reset your current landing zone, or update it to the latest version.

OU errors

- **Exceeds maximum number of SCPs**

You may be over the limit for service control policies (SCPs) per OU, or you may have reached another quota. A limit of 5 SCPs per OU applies to all OUs in your AWS Control Tower landing zone. If you have more SCPs than the quota allows, you must delete or combine the SCPs.

- **Conflicting SCPs**

  Existing SCPs may be applied to the OU or account, which prevent AWS Control Tower from enrolling the account. Check the applied SCPs for any policy that may prevent AWS Control Tower from working. Be sure to check the SCPs that are inherited from OUs higher in the hierarchy.

- **Exceeds stack set quota**

  The stack set quota may have been exceeded. If you have more instances than the quota allows, you must delete some stack instances. For more information, see AWS CloudFormation quotas in the *AWS CloudFormation User Guide*.

- **Exceeds account limit**

  AWS Control Tower limits each OU to 1000 accounts during registration.

Account errors

- **Pre-checks prevented on accounts**

  An existing SCP on the OU prevents AWS Control Tower from conducting pre-checks on your OU member accounts. To resolve this pre-check failure, update or remove the SCP from the OU.

- **Email address error**

  The email address you specified for the account does not conform to the naming standards. Here is the regular expression (regex) that specifies which characters are allowed: `[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+[.]+[A-Za-z]+`

- **Config recorder or delivery channel enabled**

  The account may have an existing AWS Config configuration recorder or delivery channel. These must be deleted or modified through the AWS CLI in all AWS Regions where the AWS Control Tower management account has governed resources, before you can enroll an account.

- **STS disabled**

AWS Security Token Service (AWS STS) may be disabled in the account. AWS STS endpoints must be activated in the accounts for all Regions supported by AWS Control Tower.

- **IAM Identity Center conflict**

  The AWS Control Tower home Region is not the same as the AWS IAM Identity Center (IAM Identity Center) Region. If IAM Identity Center is already set up, the AWS Control Tower home region must be the same as the IAM Identity Center Region.

- **Conflicting SNS topic**

  The account has an Amazon Simple Notification Service (Amazon SNS) topic name that AWS Control Tower needs to use. AWS Control Tower creates resources (such as SNS topics) with specific names. If these names are already taken, AWS Control Tower setup fails. This situation could occur if you are reusing an account previously enrolled in AWS Control Tower.

- **Suspended account detected**

  This account has been suspended. It cannot be enrolled into AWS Control Tower. Remove the account from this OU, and try again.

- **IAM user not in portfolio**

  Add the AWS Identity and Access Management (IAM) user to the Service Catalog portfolio before registering your OU. This error pertains to the management account only.

- **Account does not meet prerequisites**

  The account doesn't meet prerequisites for account enrollment. For example, the account may be missing roles and permissions required to enroll it in AWS Control Tower. Instructions for adding a role are available in Manually add the required IAM role to an existing AWS account and enroll it.

As a reminder, AWS CloudTrail is auto-enabled on all of your AWS accounts when you enroll them in AWS Control Tower. If CloudTrail is enabled on an account previous to enrollment, you could experience double-billing unless you deactivate CloudTrail before you begin the enrollment process.

Update organizations

The quickest way to update an organizational unit (OU) or to update multiple accounts within an OU is to **Re-register** the OU.

When to update AWS Control Tower OUs and accounts

When you perform a landing zone update, you must update your enrolled accounts to apply new controls to those accounts.

- You can perform an update to all accounts under an OU using the **Re-Register** option.
- If you have more than one registered OU in your landing zone, re-register all of your OUs to update all of your accounts.
- To update a single account, you can update from the AWS Control Tower console, or you can select the **Update provisioned product** option in AWS Service Catalog. See [Update the account in the console](#).

Update multiple accounts in the same OU

Repeat these steps for each OU in your AWS Control Tower organization, if you need to update all of your accounts and OUs.

To update multiple accounts in one OU, with one action

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. In the left-pane navigation menu, choose **Organization** .
3. On the **Organization** page, choose any OU to view the **OU details** page.
4. Under **Actions** in the upper right, select **Re-Register OU**.

Alternatively, you can select any account that shows a status of **Update available** and then choose **Update account**, for as many accounts as needed.

What happens during re-registration

When you re-register an OU:

- The **State** field indicates whether the account currently is enrolled with AWS Control Tower (**Enrolled**), whether the account has never been enrolled (**Not enrolled**), or whether enrollment failed previously (**Enrollment failed**).
- When you re-register the OU, the `AWSControlTowerExecution` role is added to all accounts with status **Not enrolled** or **Enrollment failed**.
- AWS Control Tower creates a single sign-on (IAM Identity Center) login for those new enrolled accounts.
- **Enrolled** accounts are re-enrolled into AWS Control Tower.
- Drift on any preventive controls applied to the OU is fixed, because the SCPs are returned to their default definitions.
- All accounts are updated to reflect the latest landing zone changes.

For more information, see Enroll an existing AWS account.

> **Tip**
>
> When you re-register an OU, or when you're updating your landing zone version and multiple member accounts, you may see a failure message mentioning the **StackSet-AWSControlTowerExecutionRole**. This StackSet in the management account can fail because the **AWSControlTowerExecution** IAM role already exists in all enrolled member accounts. This error message is expected behavior, and it can be disregarded.

Update a single account

You can update individual AWS Control Tower accounts in the AWS Control Tower console, or in the Service Catalog console.

To update a single account in the AWS Control Tower console, see Update the account in the console.

To update a single account in AWS Service Catalog

1. Go to AWS Service Catalog.
2. In the left-pane navigation menu, choose **Provisioned products**.
3. On the **Provisioned products** page, select the radio button next to the provisioned product you want to update.
4. In the upper right, choose the **Actions** dropdown to **Update**.

To learn more about updating in AWS Service Catalog, see Update the provisioned product and Updating products in the *Service Catalog Administrator Guide*.

Integrated services

AWS Control Tower is a service that's built on top of other AWS services, to assist you in setting up a well-architected environment. This chapter provides a brief overview of these services, including configuration information about the underlying services and how they work in AWS Control Tower.

For more information about how to measure a well-architected environment, learn about the AWS Well-Architected Tool. Also see the Management and Governance Cloud Environment Guide.

Topics

AWS Backup options available

AWS Backup allows you to create a backup plan for your AWS Control Tower landing zone. You can incorporate a data backup and recovery workflow directly into your landing zone. The backup plan includes predefined rules, such as retention days, backup frequency, and the time window during which backup occurs. For more information, see AWS Backup and AWS Control Tower.

Deploy Environments with AWS CloudFormation

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure. AWS CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit (a stack). For more information, see *AWS CloudFormation User Guide*.

AWS Control Tower uses AWS CloudFormation stacksets to apply controls on accounts. For more information about how AWS CloudFormation and AWS Control Tower work together, see Create AWS Control Tower resources with AWS CloudFormation.

Monitor Events with CloudTrail

AWS Control Tower configures AWS CloudTrail to enable centralized logging and auditing. With CloudTrail, the management account can review administrative actions and lifecycle events for member accounts.

CloudTrail helps you monitor your AWS environment in the cloud by keeping a history of AWS API calls for your accounts. For example, you can identify the users and accounts that called AWS APIs for services that support CloudTrail, the source IP address from which the calls were made, and the time when the calls occurred. You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of your trails, and control how administrators turn CloudTrail logging on and off. For more information, see *AWS CloudTrail User Guide*.

Monitor Resources and Services with CloudWatch

Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, and scale your own monitoring systems and infrastructure. For more information, see *Amazon CloudWatch User Guide*.

For more information about how Amazon CloudWatch works with AWS Control Tower, see Monitoring.

Govern Resource Configurations with AWS Config

AWS Config provides a detailed view of the resources associated with your AWS account, including how they are configured, how they are related to one another, and how the configurations and their relationships have changed over time. For more information, see *AWS Config Developer Guide*.

AWS Config resources provisioned by AWS Control Tower are tagged automatically with `aws-control-tower` and a value of `managed-by-control-tower`.

For more information about how AWS Config monitors and records resources in AWS Control Tower, and how it bills you for them, see Monitor resource changes with AWS Config.

AWS Control Tower uses AWS Config Rules to implement detective controls. For more information, see About controls in AWS Control Tower.

Manage Permissions for Entities with IAM

AWS Identity and Access Management (IAM) is an AWS service for controlling access to other AWS services. With IAM, you can centrally manage users, security credentials—such as access keys, and permissions—that designate the AWS resources to which your users and applications are granted access.

When you set up your landing zone, a number of groups can be created for AWS IAM Identity Center automatically, if you select IAM as your identity provider. These groups have permission sets that are pre-defined permissions policies from IAM. Your end-users also can use IAM to define the scope of permissions for IAM users and other entities within member accounts.

AWS Identity and Access Management (IAM) simplifies how you manage access to AWS accounts and business applications. You can control IAM Identity Center access and user permissions across all your AWS accounts in AWS Control Tower.

For more information, see *AWS IAM Identity Center User Guide*.

If you are based in an AWS Region that does not support IAM, you can bring another identity provider, to set up and maintain your own users and groups manually.

AWS Key Management Service

AWS Key Management Service (AWS KMS) allows you to create and control keys that protect your data. AWS Control Tower optionally allows you to encrypt your data with AWS KMS encryption keys. For information about AWS KMS, see the *AWS KMS Developer Guide*.

For information about how to set up AWS KMS keys with AWS Control Tower, see Optionally configure AWS KMS keys.

Run Serverless Compute Functions with Lambda

With AWS Lambda, you can run code without provisioning or managing servers. You can run code for many types of application or backend service— with no need for additional administration overhead. When you upload your code, Lambda can run and scale the code with high availability. You can set up your code to trigger from other AWS services automatically, or you can call it directly from any web or mobile app.

For example, certain roles in the AWS Control Tower audit account can be assumed programmatically, so that you can review other accounts using Lambda. Also, you can use AWS Control Tower lifecycle events to trigger Lambda functions.

Manage Accounts Through AWS Organizations

AWS Organizations is an account management service that lets you consolidate multiple AWS accounts into an organization that you create and centrally manage. With Organizations, you can create member accounts and invite existing accounts to join your organization. You can organize those accounts into groups and attach policy-based controls. For more information, see *AWS Organizations User Guide*.

In AWS Control Tower, Organizations helps centrally manage billing; control access, compliance, and security; and share resources across your member AWS accounts. Accounts are grouped into logical groups, called organizational units (OUs). For more information on Organizations, see *AWS Organizations User Guide*.

AWS Control Tower uses the following OUs:

- **Root** – The parent container for all accounts and all other OUs in your landing zone.
- **Security** – This OU contains the log archive account, the audit account, and the resources they own.

- **Sandbox –** This OU is created when you set up your landing zone. It and other child OUs in your landing zone contain your member accounts. These are the accounts that your end users access to perform work on AWS resources.

> **Note**
>
> You can add additional OUs in your landing zone through the AWS Control Tower console on the **Organizational units** page.

Considerations

OUs created through AWS Control Tower can have controls applied to them. OUs created outside of AWS Control Tower cannot, by default. You can, however, register such OUs. Once you have registered an OU, you can apply controls to it and its accounts. For information on registering an OU, see Register an existing organizational unit with AWS Control Tower.

Store Objects with Amazon S3

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the simple and intuitive web interface of the AWS Management Console. For more information, see *Amazon Simple Storage Service User Guide*.

When you set up your landing zone, an Amazon S3 bucket is created in your log archive account to contain all logs across all accounts in your landing zone.

Monitor your environment with Security Hub

AWS Control Tower is integrated with AWS Security Hub by means of the Security Hub standard called **Service-Managed Standard: AWS Control Tower**. For more information, see Security Hub standard.

Provision accounts through AWS Service Catalog

AWS Service Catalog enables IT administrators to create, manage, and distribute portfolios of approved products to end users, who then have access to the products they need in a personalized portal. Typical products include servers, databases, websites, or applications that are deployed using AWS resources.

You can control the users that have access to specific products, which allows you to enforce compliance with organizational business standards, manage product lifecycles, and help users find and launch products with confidence. For more information, see *Service Catalog Administrator Guide*.

In AWS Control Tower, your central cloud administrators and your end users can provision custom accounts in your landing zone using AWS Service Catalog products, called "custom blueprints". For more information, see Step2. Create the AWS Service Catalog product.

AWS Control Tower also can make use of the Service Catalog APIs to further automate account provisioning and updating. For details, see the AWS Service Catalog Developer Guide.

Transition to the AWS Service Catalog External product type

AWS Service Catalog changed support for *Terraform Open Source* products and provisioned products to a new product type, called *External*. To learn more about this transition, review Updating existing Terraform Open Source products and provisioned products to the External product type in the *AWS Service Catalog administrator guide*.

This change effects existing accounts that you created or enrolled with AWS Control Tower account factory customization. To transition these accounts to the *External* product type, you need to make changes in both AWS Service Catalog and AWS Control Tower.

To transition to the External product type

1. Upgrade your existing Terraform Reference Engine for AWS Service Catalog to include support for both *External* and *Terraform Open Source* product types. For instructions about updating your Terraform Reference Engine, review the AWS Service Catalog GitHub Repository.
2. In AWS Service Catalog, duplicate any existing *Terraform Open Source* products (blueprints), with the duplicates using the new *External* product type. **Do not terminate the existing Terraform Open Source blueprints**.
3. In AWS Control Tower, update each account using a *Terraform Open Source* blueprint to use the new *External* blueprint.

    a. To update a blueprint, you must first remove the *Terraform Open Source* blueprint completely. For more details, review Remove a blueprint from an account.

   b. Add the new *External* blueprint to the same account. For more details, review [Add a blueprint to an AWS Control Tower account].

4. After all accounts using *Terraform Open Source* blueprints are updated to *External* blueprints, return to AWS Service Catalog and terminate any products that use *Terraform Open Source* as the product type.

5. Going forward, all accounts created or enrolled using AWS Control Tower account factory customization must reference blueprints using the *AWS CloudFormation* or *External* product type.

   For blueprints created using the *External* product type, AWS Control Tower only supports account customizations that use Terraform templates and the Terraform reference engine. To learn more, review Set up for customization.

> **Note**
>
> AWS Control Tower does not support *Terraform Open Source* as a product type when creating new accounts. To learn more about these changes, review Updating existing Terraform Open Source products and provisioned products to the *External* product type in the *AWS Service Catalog administrator guide*. AWS Service Catalog will support customers through this product type transition, as needed. Contact your account representative to request assistance.

Track Alerts Through Amazon Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) is a web service that enables applications, end-users, and devices to send and receive notifications instantly from the cloud. For more information, see *Amazon Simple Notification Service Developer Guide*.

AWS Control Tower uses Amazon SNS to send programmatic alerts to the email addresses of your management account and your audit account. These alerts help you prevent drift within your landing zone. For more information, see Detect and resolve drift in AWS Control Tower.

We also use Amazon Simple Notification Service to send compliance notifications from AWS Config.

Tip

One of the best ways to receive AWS Control Tower control compliance notifications (in your audit account) is to subscribe to `AggregateConfigurationNotifications`. It is a service that helps you inspect compliance. It gives you real data about AWS Config rules going out of compliance. AWS Config automatically maintains the list of accounts in your OU.

You must subscribe manually, using email or any type of subscription that SNS allows. The statement `arn:aws:sns:`*`homeregion`*`:`*`account`*`:aws-controltower-AggregateSecurityNotifications` leads to your audit account.

Build Distributed Applications with AWS Step Functions

AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow. You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion. For more information, see *AWS Step Functions Developer Guide*.

Identity and access management in AWS Control Tower

To perform any operation in your landing zone, such as provisioning accounts in Account Factory or creating new organizational units (OUs) in the AWS Control Tower console, either AWS Identity and Access Management (IAM) or AWS IAM Identity Center require you to authenticate that you're an approved AWS user. For example, if you're using the AWS Control Tower console, you authenticate your identity by providing your AWS credentials, as provided by your administrator.

After you authenticate your identity, IAM controls your access to AWS with a defined set of permissions on a specific set of operations and resources. If you are an account administrator, you can use IAM to control the access of other IAM users to the resources that are associated with your account.

Topics

Authentication

You have access to AWS as any of the following types of identities:

- **AWS account root user –** When you first create an AWS account, you begin with an identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user. You have access to this identity when you sign in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM Identity Center user (recommended) or IAM user (not a best practice in most use cases). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. For more information, see When to sign in as a root user.

- **IAM user –** An IAM user is an identity within your AWS account that has specific, customized permissions. You can use the IAM user credentials to sign in to secure AWS webpages such as the AWS Management Console, AWS Discussion Forums, or the AWS Support Center. AWS best practices recommend that you create an IAM Identity Center user instead of an IAM user, because there is more security risk when you create an IAM user that has long-term credentials.

  If you must create an IAM user for a certain purpose, in addition to sign-in credentials, you can generate access keys for each IAM user. You can use these keys when you call AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Control Tower supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.
- **IAM role –** An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity, and it has permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:
  - **Federated user access –** Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the *IAM User Guide*.
  - **AWS service access –** A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data

from that bucket into an Amazon Redshift cluster. For more information, see [Creating a Role to Delegate Permissions to an AWS Service] in the *IAM User Guide*.

- **Applications running on Amazon EC2 –** You can use an IAM role to manage temporary credentials for applications that are running on an Amazon EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the Amazon EC2 instance. To assign an AWS role to an Amazon EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the Amazon EC2 instance to get temporary credentials. For more information, see [Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances] in the *IAM User Guide*.

- **IAM Identity Center user** Authentication to the IAM Identity Center user portal is controlled by the directory that you have connected to IAM Identity Center. However, authorization to the AWS accounts that are available to end users from within the user portal is determined by two factors:
  - Who has been assigned access to those AWS accounts in the AWS IAM Identity Center console. For more information, see [Single Sign-On Access] in the *AWS IAM Identity Center User Guide*.
  - What level of permissions have been granted to the end-users in the AWS IAM Identity Center console to allow them the appropriate access to those AWS accounts. For more information, see [Permission Sets] in the *AWS IAM Identity Center User Guide*.

Access control

To create, update, delete, or list AWS Control Tower resources, or other AWS resources in your landing zone you need permissions to perform the operation, and you need permissions to access the corresponding resources. In addition, to perform the operation programmatically, you need valid access keys.

The following sections describe how to manage permissions for AWS Control Tower:

Topics

Working with AWS IAM Identity Center and AWS Control Tower

In AWS Control Tower, IAM Identity Center allows central cloud administrators and end-users to manage access to multiple AWS accounts and business applications. By default, AWS Control Tower uses this service to set up and manage access to the accounts created through Account Factory, unless you have selected the option to self-manage your identity and access control.

For more information about selecting an identity provder, see [IAM Identity Center guidance](#).

For a brief tutorial about how to set up your IAM Identity Center users and permissions in AWS Control Tower, you can view this video (6:23). For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

[Video Walkthrough of Setting Up AWS IAM Identity Center in AWS Control Tower.](#)

About setting up AWS Control Tower with IAM Identity Center

When you initially set up AWS Control Tower, only the root user user and any IAM users with the correct permissions can add IAM Identity Center users. However, after end users have been added in the **AWSAccountFactory** group, they can create new IAM Identity Center users from the Account Factory wizard. For more information, see [Provision and manage accounts with Account Factory](#).

If you choose the recommended default, AWS Control Tower sets up your landing zone with a preconfigured directory that helps you manage user identities and single sign-on, so that your users have federated access across accounts. When you set up your landing zone, this default directory is created to contain *user groups* and *permission sets*.

> **Note**
>
> You can delegate administration of AWS IAM Identity Center in your organization to an account other than the management account, by using the delegated administrator feature of IAM Identity Center. If you choose to use this feature, be aware that Administrators with access to manage group membership *also* can manage groups assigned to the management account. For more information, see this blog post, entitled, [Getting started with AWS SSO delegated administration](#)

User groups, roles, and permission sets

User groups manage specialized *roles* that are defined within your shared accounts. Roles establish sets of permissions that belong together. All members of a group inherit the permission sets, or roles, associated with the group. You can create new groups for the end users of your member accounts, so that you can custom-assign only the roles that are needed for the specific tasks a group performs.

The permission sets available cover a broad range of distinct user permission requirements, such as read-only access, AWS Control Tower administrative access, and Service Catalog access. These permission sets enable your end users to provision their own AWS accounts in your landing zone quickly, and in compliance with your enterprise's guidelines.

For tips on planning your allocations of users, groups, and permissions, refer to Recommendations for setting up groups, roles, and policies

For more information on how to use this service in the context of AWS Control Tower, see the following topics in the *AWS IAM Identity Center User Guide*.

- To add users, see Add Users.
- To add users to groups, see Add Users to Groups.
- To edit user properties, see Edit User Properties.
- To add a group, see Add Groups.

> **Warning**
>
> AWS Control Tower sets up your IAM Identity Center directory in your home Region. If you set up your landing zone in another Region and then navigate to the IAM Identity Center console, you must change the Region to your home region. Do not delete your IAM Identity Center configuration in your home Region.

Things to know about IAM Identity Center accounts and AWS Control Tower

Here are some good things to know when working with IAM Identity Center user accounts in AWS Control Tower.

- If your AWS IAM Identity Center user account is disabled, you'll get an error message when trying to provision new accounts in Account Factory. You can re-enable your IAM Identity Center user in the IAM Identity Center console.
- If you specify a new IAM Identity Center user email address when you update the provisioned product associated with an account that was vended by Account Factory, AWS Control Tower creates a new IAM Identity Center user account. The previously created user account is not removed. If you prefer to remove the previous IAM Identity Center user email address from AWS IAM Identity Center, see Disabling a User.
- AWS IAM Identity Center has been integrated with Azure Active Directory, and you can connect your existing Azure Active Directory to AWS Control Tower.
- For more information about how the behavior of AWS Control Tower interacts with AWS IAM Identity Center and different identity sources, refer to the Considerations for Changing Your Identity Source in the AWS IAM Identity Center documentation.

IAM Identity Center Groups for AWS Control Tower

AWS Control Tower offers preconfigured groups to organize users that perform specific tasks in your accounts. You can add users and assign them to these groups directly in IAM Identity Center. Doing so matches permission sets to users in groups within your accounts. For the latest guidance and best practices on configuring your groups, see Best practices in the *IAM Identity Center User Guide*.

The following groups are created when you set up your landing zone.

AWSAccountFactory

Account	Permission sets	Description
Management account	AWSServiceCatalogEndUserAccess	This group is only used in this account to provision new accounts using Account Factory.

AWSServiceCatalogAdmins

Account	Permission sets	Description
Management account	AWSServiceCatalogA dminFullAccess	This group is only used in this account to make administr ative changes to Account Factory. Users in this group can't provision new accounts unless they're also in the **AWSAccountFactory** group.

AWSControlTowerAdmins

Account	Permission sets	Description
Management account	AWSAdministratorAccess	Users of this group in this account are the only ones that have access to the AWS Control Tower console.
Log archive account	AWSAdministratorAccess	Users have administrator access in this account.
Audit account	AWSAdministratorAccess	Users have administrator access in this account.
Member accounts	AWSOrganizationsFullAccess	Users have full access to Organizations in this account.

AWSSecurityAuditPowerUsers

Account	Permission sets	Description
Management account	AWSPowerUserAccess	Users can perform applicati on development tasks and can create and configure resources and services that

Account	Permission sets	Description
		support AWS aware applicati on development.
Log archive account	AWSPowerUserAccess	Users can perform applicati on development tasks and can create and configure resources and services that support AWS aware applicati on development.
Audit account	AWSPowerUserAccess	Users can perform applicati on development tasks and can create and configure resources and services that support AWS aware applicati on development.
Member accounts	AWSPowerUserAccess	Users can perform applicati on development tasks and can create and configure resources and services that support AWS aware applicati on development.

AWSSecurityAuditors

Account	Permission sets	Description
Management account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.
Log archive account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

Account	Permission sets	Description
Audit account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.
Member accounts	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

AWSLogArchiveAdmins

Account	Permission sets	Description
Log archive account	AWSAdministratorAccess	Users have administrator access in this account.

AWSLogArchiveViewers

Account	Permission sets	Description
Log archive account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

AWSAuditAccountAdmins

Account	Permission sets	Description
Audit account	AWSAdministratorAccess	Users have administrator access in this account.

Overview of managing access permissions to your AWS Control Tower resources

Every AWS resource is owned by an AWS account, and permissions to create or gain access to a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
>
> An *account administrator* (or administrator) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When you are responsible for granting permissions to a user or role, you must know and track the *users and roles* that require permissions, the *resources* for which each user and role requires permissions, and the *specific actions* that must be allowed for operating those resources.

Topics

AWS Control Tower resources and operations

In AWS Control Tower, the primary resource is a *landing zone*. AWS Control Tower also supports an additional resource type, *controls*, sometimes referred to as *guardrails*. However, for AWS Control Tower, you can manage controls only in the context of an existing landing zone. Controls can be referred to as a *subresource*.

Resources and subresources in AWS have unique Amazon Resource Names (ARNs) associated with them, as shown in the following example.

AWS Control Tower provides a set of API operations to work with AWS Control Tower resources. For a list of available operations, see AWS Control Tower the AWS Control Tower API Reference.

For more information about the AWS CloudFormation resources in AWS Control Tower, see the AWS CloudFormation User Guide.

About resource ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the AWS account root user, an IAM Identity Center user, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the AWS account root user credentials of your AWS account to set up a landing zone, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to set up a landing zone to that user, the user can set up a landing zone as long as their account meets the prerequisites. However, your AWS account, to which the user belongs, owns the landing zone resource.
- If you create an IAM role in your AWS account with permissions to set up a landing zone, anyone who can assume the role can set up a landing zone. Your AWS account, to which the role belongs, owns the landing zone resource.

Manage access to resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
>
> This section discusses using IAM in the context of AWS Control Tower. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices). Policies attached to a resource are referred to as *resource-based* policies.

> **Note**
>
> AWS Control Tower supports only identity-based policies (IAM policies).

Topics

About identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create an AWS Control Tower resource, such as setting up a landing zone, you can attach a permissions policy to a user or group that the user belongs to.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, an administrator for one AWS account (*Account A*) can create a role that grants cross-account permissions to another AWS account (*Account B*), or the administrator can create a role that grants permissions to another AWS service.
  1. The Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions to manage resources in Account A.
  2. The Account A administrator attaches a trust policy to the role. The policy identifies Account B as the principal who can assume the role.
  3. As principal, the Account B administrator can give any user in Account B permission to assume the role. By assuming the role, users in Account B can create or gain access to resources in Account A.
  4. To grant an AWS service the ability (permissions) to assume the role, the principal that you specify in the trust policy can be an AWS service.

Create roles and assign permissions

Roles and permissions give you access to resources, in AWS Control Tower and in other AWS services, including programmatic access to resources.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

> **Note**
>
> When setting up an AWS Control Tower landing zone, you'll need a user or role with the **AdministratorAccess** managed policy. (arn:aws:iam::aws:policy/AdministratorAccess)

To create a role for an AWS service (IAM console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create role**.
3. For **Trusted entity type**, choose **AWS service**.

4. For **Service or use case**, choose a service, and then choose the use case. Use cases are defined by the service to include the trust policy that the service requires.
5. Choose **Next**.
6. For **Permissions policies**, the options depend on the use case that you selected:
   - If the service defines the permissions for the role, you can't select permissions policies.
   - Select from a limited set of permission polices.
   - Select from all permission policies.
   - Select no permissions policies, create the policies after the role is created, and then attach the policies to the role.
7. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not service-linked roles.
   a. Open the **Set permissions boundary** section, and then choose **Use a permissions boundary to control the maximum role permissions**.

      IAM includes a list of the AWS managed and customer-managed policies in your account.
   b. Select the policy to use for the permissions boundary.
8. Choose **Next**.
9. For **Role name**, the options depend on the service:
   - If the service defines the role name, you can't edit the role name.
   - If the service defines a prefix for the role name, you can enter an optional suffix.
   - If the service doesn't define the role name, you can name the role.

> **Important**
>
> When you name a role, note the following:
>
> - Role names must be unique within your AWS account, and can't be made unique by case.
>
>   For example, don't create roles named both `PRODROLE` and `prodrole`. When a role name is used in a policy or as part of an ARN, the role name is case sensitive, however when a role name appears to customers in the console, such as during the sign-in process, the role name is case insensitive.

- You can't edit the name of the role after it's created because other entities might reference the role.

10. (Optional) For **Description**, enter a description for the role.
11. (Optional) To edit the use cases and permissions for the role, in the **Step 1: Select trusted entities** or **Step 2: Add permissions** sections, choose **Edit**.
12. (Optional) To help identify, organize, or search for the role, add tags as key-value pairs. For more information about using tags in IAM, see Tags for AWS Identity and Access Management resources in the *IAM User Guide*.
13. Review the role, and then choose **Create role**.

To use the JSON policy editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**.

   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
3. At the top of the page, choose **Create policy**.
4. In the **Policy editor** section, choose the **JSON** option.
5. Enter or paste a JSON policy document. For details about the IAM policy language, see IAM JSON policy reference.
6. Resolve any security warnings, errors, or general warnings generated during policy validation, and then choose **Next**.

   Note

   You can switch between the **Visual** and **JSON** editor options anytime. However, if you make changes or choose **Next** in the **Visual** editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the *IAM User Guide*.
7. (Optional) When you create or edit a policy in the AWS Management Console, you can generate a JSON or YAML policy template that you can use in AWS CloudFormation templates.

To do this, in the **Policy editor** choose **Actions**, and then choose **Generate CloudFormation template**. To learn more about AWS CloudFormation, see [AWS Identity and Access Management resource type reference] in the *AWS CloudFormation User Guide*.

8. When you are finished adding permissions to the policy, choose **Next**.
9. On the **Review and create** page, enter a **Policy name** and a **Description** (optional) for the policy that you are creating. Review **Permissions defined in this policy** to see the permissions that are granted by your policy.
10. (Optional) Add metadata to the policy by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tags for AWS Identity and Access Management resources] in the *IAM User Guide*.
11. Choose **Create policy** to save your new policy.

To use the visual editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**.

   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
3. Choose **Create policy**.
4. In the **Policy editor** section, find the **Select a service** section, and then choose an AWS service. You can use the search box at the top to limit the results in the list of services. You can choose only one service within a visual editor permission block. To grant access to more than one service, add multiple permission blocks by choosing **Add more permissions**.
5. In **Actions allowed**, choose the actions to add to the policy. You can choose actions in the following ways:

   - Select the check box for all actions.
   - Choose **Add actions** to enter the name of a specific action. You can use a wildcard character (*) to specify multiple actions.
   - Select one of the **Access level** groups to choose all actions for the access level (for example, **Read**, **Write**, or **List**).
   - Expand each of the **Access level** groups to choose individual actions.

By default, the policy that you are creating allows the actions that you choose. To deny the chosen actions instead, choose **Switch to deny permissions**. Because IAM denies by default, we recommend as a security best practice that you allow permissions to only those actions and resources that a user needs. Create a JSON statement to deny permissions only if you want to override a permission separately allowed by another statement or policy. We recommend that you limit the number of deny permissions to a minimum because they can increase the difficulty of troubleshooting permissions.

6. For **Resources**, if the service and actions that you selected in the previous steps do not support choosing specific resources, all resources are allowed and you cannot edit this section.

   If you chose one or more actions that support resource-level permissions, then the visual editor lists those resources. You can then expand **Resources** to specify resources for your policy.

   You can specify resources in the following ways:

   - Choose **Add ARNs** to specify resources by their Amazon Resource Names (ARN). You can use the visual ARN editor or list ARNs manually. For more information about ARN syntax, see Amazon Resource Names (ARNs) in the *IAM User Guide*. For information about using ARNs in the `Resource` element of a policy, see IAM JSON policy elements: Resource in the *IAM User Guide*.
   - Choose **Any in this account** next to a resource to grant permissions to any resources of that type.
   - Choose **All** to choose all resources for the service.

7. (Optional) Choose **Request conditions - *optional*** to add conditions to the policy that you are creating. Conditions limit a JSON policy statement's effect. For example, you can specify that a user is allowed to perform the actions on the resources only when that user's request happens within a certain time range. You can also use commonly used conditions to limit whether a user must be authenticated by using a multi-factor authentication (MFA) device. Or you can require that the request originate from within a certain range of IP addresses. For lists of all of the context keys that you can use in a policy condition, see Actions, resources, and condition keys for AWS services in the *Service Authorization Reference*.

   You can choose conditions in the following ways:

   - Use check boxes to select commonly used conditions.

- Choose **Add another condition** to specify other conditions. Choose the condition's **Condition Key**, **Qualifier**, and **Operator**, and then enter a **Value**. To add more than one value, choose **Add**. You can consider the values as being connected by a logical OR operator. When you are finished, choose **Add condition**.

   To add more than one condition, choose **Add another condition** again. Repeat as needed. Each condition applies only to this one visual editor permission block. All the conditions must be true for the permission block to be considered a match. In other words, consider the conditions to be connected by a logical AND operator.

   For more information about the **Condition** element, see [IAM JSON policy elements: Condition] in the *IAM User Guide*.

8. To add more permission blocks, choose **Add more permissions**. For each block, repeat steps 2 through 5.

   > **Note**
   >
   > You can switch between the **Visual** and **JSON** editor options anytime. However, if you make changes or choose **Next** in the **Visual** editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see [Policy restructuring] in the *IAM User Guide*.

9. (Optional) When you create or edit a policy in the AWS Management Console, you can generate a JSON or YAML policy template that you can use in AWS CloudFormation templates.

   To do this, in the **Policy editor** choose **Actions**, and then choose **Generate CloudFormation template**. To learn more about AWS CloudFormation, see [AWS Identity and Access Management resource type reference] in the *AWS CloudFormation User Guide*.

10. When you are finished adding permissions to the policy, choose **Next**.

11. On the **Review and create** page, enter a **Policy name** and a **Description** (optional) for the policy that you are creating. Review the **Permissions defined in this policy** to make sure that you have granted the intended permissions.

12. (Optional) Add metadata to the policy by attaching tags as key-value pairs. For more information about using tags in IAM, see [Tags for AWS Identity and Access Management resources] in the *IAM User Guide*.

13. Choose **Create policy** to save your new policy.

To grant programmatic access

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
Workforce identity (Users managed in IAM Identity Center)	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*. • For AWS SDKs, tools, and AWS APIs, see IAM Identity Center authentication in the *AWS SDKs and Tools Reference Guide*.
IAM	Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions in Using temporary credentials with AWS resources in the *IAM User Guide*.
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.	Following the instructions for the interface that you want to use. • For the AWS CLI, see Authenticating using IAM user credentials in the *AWS*

Which user needs programmatic access?	To	By
		Command Line Interface User Guide. • For AWS SDKs and tools, see Authenticate using long-term credentials in the *AWS SDKs and Tools Reference Guide*. • For AWS APIs, see Managing access keys for IAM users in the *IAM User Guide*.

Protect against attackers

For more information about how to help protect against attackers when you grant permissions to other AWS service principals, see Optional conditions for your role trust relationships. By adding certain conditions to your policies, you can help prevent a specific type of attack, known as a *confused deputy* attack, which occurs if an entity coerces a more-privileged entity to perform an action, such as with cross-service impersonation. For general information about policy conditions, also see Specifying conditions in a policy.

For more information about using identity-based policies with AWS Control Tower, see Using identity-based policies (IAM policies) for AWS Control Tower. For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Control Tower does not support resource-based policies.

Specify policy elements: Actions, Effects, and Principals

You can set up and manage your landing zone through the AWS Control Tower console, or the landing zone APIs. To set up your landing zone, you must be an IAM user with administrative permissions as defined in a IAM policy.

The following elements are the most basic ones you can identify in a policy:

- **Resource –** In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS Control Tower resources and operations.
- **Action –** You use action keywords to identify resource operations that you want to allow or deny. For information about types of actions available to be performed, see Actions defined by AWS Control Tower.
- **Effect –** You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal –** In identity-based policies (IAM policies), that user to which the policy is attached is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Control Tower doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Specifying conditions in a policy

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you can use predefined condition keys. There are no condition keys specific to AWS Control Tower. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the *IAM User Guide*.

Prevent cross-service impersonation

In AWS, cross-service impersonation can result in the *confused deputy problem*. When one service calls another service, cross-service impersonation occurs if one service manipulates another service to use its permissions to act on a customer's resources in a way that's not otherwise permitted. To prevent this attack, AWS provides tools to help you protect your data, so that only those services with legitimate permission can gain access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` conditions in your policies, to limit the permissions that AWS Control Tower gives to another service for access to your resources.

- Use `aws:SourceArn` if you want only one resource to be associated with cross-service access.
- Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with cross-service use.
- If the `aws:SourceArn` value does not contain the account ID, such as the ARN for an Amazon S3 bucket, you must use both conditions to limit permissions.
- If you use both conditions, and if the `aws:SourceArn` value contains the account ID, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must show the same account ID when used in the same policy statement

For more information and examples, see https://docs.aws.amazon.com/controltower/latest/userguide/conditions-for-role-trust.html.

Using identity-based policies (IAM policies) for AWS Control Tower

This topic provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Control Tower resources.

> **Important**
>
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Control Tower resources. For

more information, see Overview of managing access permissions to your AWS Control Tower resources.

Permissions Required to use the AWS Control Tower console

AWS Control Tower creates three roles automatically when you set up a landing zone. All three roles are required to allow console access. AWS Control Tower splits permissions into three roles as a best practice to restrict access to the minimal sets of actions and resources.

Three required roles for landing zone access

- AWSControlTowerAdmin role
- AWSControlTowerStackSetRole
- AWSControlTowerCloudTrailRole

We recommend that you restrict access to your role trust policies for these roles. For more information, see Optional conditions for your role trust relationships.

View the Control Catalog in the console

To view control information in the AWS Control Tower console, you must add additional `controlcatalog` permissions to your IAM policies. These permissions are as follows:

- `controlcatalog:GetControl`
- `controlcatalog:ListControls`
- `controlcatalog:ListControlMappings`
- `controlcatalog:ListCommonControls`

Here's an example showing the updated permissions in the policy.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Action": [
                "controlcatalog:GetControl",
                "controlcatalog:ListControls",
                "controlcatalog:ListControlMappings",
                "controlcatalog:ListCommonControls"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

You must add these permissions because AWS Control Tower calls the `controlcatalog` APIs to retrieve certain control metadata, so the AWS Control Tower permissions are not sufficient.

To find more information about how to update your permissions, see Create roles and assign permissions.

To find more information about `controlcatalog` IAM actions, see Actions, resources, and condition keys for Control Catalog.

Note

Control information is available through the Control Catalog APIs.

AWSControlTowerAdmin role

This role provides AWS Control Tower with access to infrastructure critical to maintaining the landing zone. The `AWSControlTowerAdmin` role requires an attached managed policy and a role trust policy for the IAM role. A *role trust policy* is a resource-based policy, specifying which principals can assume the role.

Here's an example snippet for this role trust policy:

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "controltower.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

To create this role from the AWS CLI, and put it into a file called `trust.json`, here's an example CLI command:

```
aws iam create-role --role-name AWSControlTowerAdmin --path /service-role/ --assume-role-policy-document file://trust.json
```

This role requires two IAM policies.

1. An inline policy, for example:

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "ec2:DescribeAvailabilityZones",
               "Resource": "*"
           }
       ]
   }
   ```

2. The managed policy that follows, which is the `AWSControlTowerServiceRolePolicy`.

AWSControlTowerServiceRolePolicy

The **AWSControlTowerServiceRolePolicy** is an AWS-managed policy that defines permissions to create and manage AWS Control Tower resources, such as AWS CloudFormation stacksets and stack instances, AWS CloudTrail log files, a configuration aggregator for AWS Control Tower, as well as AWS Organizations accounts and organizational units (OUs) that are governed by AWS Control Tower.

Updates to this managed policy are summarized in the table, [Managed policies for AWS Control Tower](#).

For more information, see [AWSControlTowerServiceRolePolicy](#) in the *AWS Managed Policy Reference Guide*.

Role trust policy:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "controltower.amazonaws.com"
         ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The inline policy is `AWSControlTowerAdminPolicy`:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*",
```

```
            "Effect": "Allow"
        }
    ]
}
```

AWSControlTowerStackSetRole

AWS CloudFormation assumes this role to deploy stack sets in accounts created by AWS Control Tower. Inline Policy:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWSControlTowerExecution"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Trust policy

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
```

AWSControlTowerCloudTrailRole

AWS Control Tower enables CloudTrail as a best practice and provides this role to CloudTrail. CloudTrail assumes this role to create and publish CloudTrail logs. Inline Policy:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "logs:CreateLogStream",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/
CloudTrailLogs:*",
            "Effect": "Allow"
        },
        {
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/
CloudTrailLogs:*",
            "Effect": "Allow"
        }
    ]
}
```

Trust policy

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

AWSControlTowerBlueprintAccess role requirements

AWS Control Tower requires you to create the `AWSControlTowerBlueprintAccess` role in the designated blueprint hub account, within the same organization.

Role name

The role name must be `AWSControlTowerBlueprintAccess`.

Role trust policy

The role must be set up to trust the following principals:

- The principal that uses AWS Control Tower in the management account.
- The `AWSControlTowerAdmin` role in the management account.

The following example shows a least-privilege trust policy. When you make your own policy, replace the term *`YourManagementAccountId`* with the actual acccount ID of your AWS Control Tower management account, and replace the term *`YourControlTowerUserRole`* with the identifier of the IAM role for your management account.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/service-role/
AWSControlTowerAdmin",
```

```
                    "arn:aws:iam::111122223333:role/YourControlTowerUserRole"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

Role permissions

You are required to attach the managed policy **AWSServiceCatalogAdminFullAccess** to the role.

AWSServiceRoleForAWSControlTower

This role provides AWS Control Tower with access to the Log Archive account, Audit account, and member accounts, for operations critical to maintaining the landing zone, such as notifying you of drifted resources.

The `AWSServiceRoleForAWSControlTower` role requires an attached managed policy and a role trust policy for the IAM role.

Managed policy for this role: `AWSControlTowerAccountServiceRolePolicy`

Role trust policy:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "controltower.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

AWSControlTowerAccountServiceRolePolicy

This AWS-managed policy allows AWS Control Tower to call AWS services that provide automated account configuration and centralized governance on your behalf.

The policy contains the minimum permissions for AWS Control Tower to implement AWS Security Hub findings forwarding for resources managed by Security Hub controls that are part of the **Security Hub Service-managed Standard: AWS Control Tower**, and it prevents changes that restrict the ability to manage customer accounts. It is part of background AWS Security Hub drift detection process that is not directly initiated by a customer.

The policy gives permissions to create Amazon EventBridge rules, specifically for Security Hub controls, in each member account, and these rules must specify an exact EventPattern. Also, a rule can operate only on rules managed by our service principal.

Service principal: `controltower.amazonaws.com`

For more information, see `AWSControlTowerAccountServiceRolePolicy` in the *AWS Managed Policy Reference Guide*.

Updates to this managed policy are summarized in the table, Managed policies for AWS Control Tower.

Managed policies for AWS Control Tower

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the *IAM User Guide*.

Change	Description	Date
AWSControlTowerServiceRolePolicy – Update to an existing policy	AWS Control Tower added new permissions that allow customers to enable and disable service-linked AWS Config rules. This change is needed so that customers can manage	June 5, 2025

Change	Description	Date
	controls that are deployed by Config rules.	
[AWSControlTowerServiceRolePolicy]() – Update to an existing policy	AWS Control Tower added new permissions that allow AWS Control Tower to make calls to the AWS CloudFormation service APIs `ActivateType` , `DeactivateType` , and `SetTypeConfiguration` , on `AWS::ControlTower` `types`. This change allows customers to provision proactive controls without the deployment of private AWS CloudFormation Hook types.	December 10, 2024

Change	Description	Date
AWSControlTowerAccountServiceRolePolicy – A new policy	AWS Control Tower added a new service-linked role that allows AWS Control Tower to create and manage event rules, and based on those rules, to manage drift detection for controls that are related to Security Hub. This change is needed so that customers can view drifted resources in the console, when those resources are related to Security Hub controls that are part of the **Security Hub Service-managed Standard: AWS Control Tower**.	May 22, 2023

Change	Description	Date
AWSControlTowerServiceRolePolicy – Update to an existing policy	AWS Control Tower added new permissions that allow AWS Control Tower to make calls to the `EnableRegion` , `ListRegions` , and `GetRegionOptStatus` APIs implemented by the AWS Account Management service, to make the opt-in AWS Regions available for customer accounts in the landing zone (Management account, Log archive account, Audit account, OU member accounts). This change is needed so that customers can have the option to expand Region governance by AWS Control Tower into the opt-in Regions.	April 6, 2023

Change	Description	Date
AWSControlTowerServiceRolePolicy – Update to an existing policy	AWS Control Tower added new permissions that allow AWS Control Tower to assume the `AWSControlTowerBlueprintAccess` role in the blueprint (hub) account, which is a dedicated account in an organization, containing pre-defined blueprints stored in one or more Service Catalog Products. AWS Control Tower assumes the `AWSControlTowerBlueprintAccess` role to perform three tasks: create a Service Catalog Portfolio, add the requested blueprint Product, and share the Portfolio to a requested member account at account provisioning time. This change is needed so that customers can provision customized accounts through AWS Control Tower Account Factory.	October 28, 2022

Change	Description	Date
AWSControlTowerServiceRolePolicy – Update to an existing policy	AWS Control Tower added new permissions that allow customers to set up organization-level AWS CloudTrail trails, starting in landing zone version 3.0. The organization-based CloudTrail feature requires customers to have trusted access enabled for the CloudTrail service, and the IAM user or role must have permission to create an organization-level trail in the management account.	June 20, 2022

Change	Description	Date
AWSControlTowerServiceRolePolicy – Update to an existing policy	AWS Control Tower added new permissions that allow customers to use KMS key encryption. The KMS feature allows customers to provide their own KMS key to encrypt their CloudTrail logs. Customers also can change the KMS key during landing zone update or repair. When updating the KMS key, AWS CloudFormation needs permissions to call the AWS CloudTrail `PutEventSelector` API. The change to the policy is to allow the **AWSControlTowerAdmin** role to call the AWS CloudTrail `PutEventSelector` API.	July 28, 2021
AWS Control Tower started tracking changes	AWS Control Tower started tracking changes for its AWS managed policies.	May 27, 2021

Security in AWS Control Tower

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Control Tower, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS services that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Control Tower. The following topics show you how to configure AWS Control Tower to meet your security and compliance objectives. You also learn how to use other AWS services that help you monitor and secure your AWS Control Tower resources.

Data Protection in AWS Control Tower

The AWS shared responsibility model applies to data protection in AWS Control Tower. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS Control Tower or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

> **Note**
>
> User activity logging with AWS CloudTrail is handled automatically in AWS Control Tower when you set up your landing zone.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*. AWS Control Tower provides the following options that you can use to help secure the content that exists in your landing zone:

Topics

- Encryption at Rest
- Encryption in Transit
- Restrict Access to Content

Encryption at Rest

AWS Control Tower uses Amazon S3 buckets and Amazon DynamoDB databases that are encrypted at rest by using Amazon S3-Managed Keys (SSE-S3) in support of your landing zone. This encryption is configured by default when you set up your landing zone. Optionally, you can configure your landing zone to encrypt resources with KMS encryption keys. You can also establish encryption at rest for the services you use in your landing zone for the services that support it. For more information, see the security chapter of that service's online documentation.

Encryption in Transit

AWS Control Tower uses Transport Layer Security (TLS) and client-side encryption for encryption in transit in support of your landing zone. In addition, accessing AWS Control Tower requires using the console, which can only be accessed through an HTTPS endpoint. This encryption is configured by default when you set up your landing zone.

Restrict Access to Content

As a best practice, you should restrict access to the appropriate subset of users. With AWS Control Tower, you can do this by ensuring that your central cloud administrators and end users have the right IAM permissions or, in the case of IAM Identity Center users, that they are in the correct groups.

- For more information about roles and policies for IAM entities, see *IAM User Guide*.
- For more information about the IAM Identity Center groups that are created when you set up your landing zone, see IAM Identity Center Groups for AWS Control Tower.

Compliance Validation for AWS Control Tower

AWS Control Tower is a well-architected service that can help your organization meet your compliance needs with controls and best practices. Additionally, third-party auditors assess the security and compliance of a number of the services you can use in your landing zone as a part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact in the *AWS Artifact User Guide*.

Your compliance responsibility when using AWS Control Tower is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Control Tower

The AWS global infrastructure is built around AWS Regions and Availability Zones.

AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected by means of low-latency, high-throughput, and highly redundant networking. Availability Zones allow you to design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For a list of AWS Regions where AWS Control Tower is available, see How AWS Regions Work With AWS Control Tower.

Your *home region* is defined as the AWS Region in which your landing zone was set up.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure Security in AWS Control Tower

AWS Control Tower is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls for access to AWS services and resources within your landing zone through the network. We require Transport Layer Security (TLS) 1.2 and recommend Transport Layer Security (TLS) 1.3 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

You can set up security groups to provide additional network infrastructure security for your AWS Control Tower landing zone workloads. For more information, see Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager.

Logging and monitoring in AWS Control Tower

Monitoring allows you to plan for and respond to potential incidents. The results of monitoring activities are stored in log files. Therefore, logging and monitoring are closely related concepts, and they are an important part of the well-architected nature of AWS Control Tower.

When you set up your landing zone, one of the shared accounts created is the *log archive* account. It is dedicated to collecting all logs centrally, including logs for all of your shared and member accounts. Log files are stored in an Amazon S3 bucket. These log files allow administrators and auditors to review actions and events that have occurred.

As a best practice, you should collect monitoring data from all of the parts of your AWS setup into your logs, so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your resources and activity in your landing zone.

For example, the status of your controls is monitored constantly. You can see their status at a glance in the AWS Control Tower console, or programmatically by means of the AWS Control Tower APIs. The health and status of the accounts you provisioned in Account Factory also is monitored constantly.

View logged actions from the Activities page

In the AWS Control Tower console, the **Activities** page provides an overview of AWS Control Tower management account actions. To navigate to the AWS Control Tower **Activities** page, select **Activities** from the left navigation.

The activities shown in the **Activities** page are the same ones reported in the AWS CloudTrail events log for AWS Control Tower, but they're shown in a table format. To learn more about a specific activity, select the activity from the table and then choose **View details**.

You can view member account actions and events in the log archive files.

The following sections describe monitoring and logging in AWS Control Tower with more detail:

Topics

About logging in AWS Control Tower

AWS Control Tower accomplishes logging of actions and events automatically, through its integration with AWS CloudTrail and AWS Config, and it records them in CloudWatch. All actions are logged, including actions from the AWS Control Tower management account and from your organization's member accounts. Management account actions and events are viewable on the **Activities** page in the console. You can view member account actions and events in the log archive files.

Organization-level trails

AWS Control Tower sets up a new CloudTrail trail when you set up a landing zone. It is an *organization-level trail*, which means that it logs all events for the management account and all member accounts in the organization. This feature relies on *trusted access* to give the management account permissions to create a trail on every member account.

For more information about AWS Control Tower and CloudTrail organization trails, see Creating a trail for an organization.

> **Note**
>
> In AWS Control Tower releases before landing zone version 3.0, AWS Control Tower created a member account trail in each account. When you update to release 3.0, your CloudTrail trail becomes an organization trail. For best practices when moving between trails, see Best practices for changing trails in the *CloudTrail User Guide*.

When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the AWS Control Tower organization. If you have an existing deployment of a CloudTrail trail in that account, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower.

> **Note**
>
> When you update to landing zone version 3.0, AWS Control Tower deletes the account-level trails (that AWS Control Tower has created) in your enrolled accounts on your behalf. Your existing, account-level log files are preserved in their Amazon S3 bucket.

Amazon S3 bucket policy in the audit account

In AWS Control Tower, AWS services have access to your resources only when the request originates from your organization or organizational unit (OU). An `aws:SourceOrgID` condition must be met for any write permissions.

You can use the `aws:SourceOrgID` condition key and set the value to your **organization ID** in the condition element of your Amazon S3 bucket policy. This condition ensures that CloudTrail only can write logs on behalf of accounts within your organization to your S3 bucket; it prevents CloudTrail logs outside your organization from writing to your AWS Control Tower S3 bucket.

This policy does not affect the functionality of your existing workloads. The policy is shown in the example that follows.

```
S3AuditBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref S3AuditBucket
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Sid: AllowSSLRequestsOnly
            Effect: Deny
            Principal: '*'
            Action: s3:*
            Resource:
             - !Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}"
             - !Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}/*"
            Condition:
              Bool:
                aws:SecureTransport: false
          - Sid: AWSBucketPermissionsCheck
            Effect: Allow
            Principal:
              Service:
                - cloudtrail.amazonaws.com
                - config.amazonaws.com
            Action: s3:GetBucketAcl
            Resource:
              - !Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}"
          - Sid: AWSConfigBucketExistenceCheck
            Effect: Allow
```

```
            Principal:
              Service:
                - cloudtrail.amazonaws.com
                - config.amazonaws.com
            Action: s3:ListBucket
            Resource:
              - !Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}"
          - Sid: AWSBucketDeliveryForConfig
            Effect: Allow
            Principal:
              Service:
                - config.amazonaws.com
            Action: s3:PutObject
            Resource:
              - Fn::Join:
                  - ""
                  -
                    - !Sub "arn:${AWS::Partition}:s3:::"
                    - !Ref "S3AuditBucket"
                    - !Sub "/${AWSLogsS3KeyPrefix}/AWSLogs/*/*"
            Condition:
              StringEquals:
                aws:SourceOrgID: !Ref OrganizationId
          - Sid: AWSBucketDeliveryForOrganizationTrail
            Effect: Allow
            Principal:
              Service:
                - cloudtrail.amazonaws.com
            Action: s3:PutObject
            Resource: !If [IsAccountLevelBucketPermissionRequiredForCloudTrail,
                [!Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}/
${AWSLogsS3KeyPrefix}/AWSLogs/${Namespace}/*", !Sub "arn:${AWS::Partition}:s3:::
${S3AuditBucket}/${AWSLogsS3KeyPrefix}/AWSLogs/${OrganizationId}/*"],
                !Sub "arn:${AWS::Partition}:s3:::${S3AuditBucket}/
${AWSLogsS3KeyPrefix}/AWSLogs/*/*"]
            Condition:
              StringEquals:
            aws:SourceOrgID: !Ref OrganizationId
```

For more information about this condition key, see the IAM documentation and the IAM blog post entitled "*Use scalable controls for AWS services accessing your resources*."

Integrated tools for monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Control Tower and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Control Tower, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *Amazon CloudWatch Events* delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the Amazon CloudWatch Events User Guide.
- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

Tip: You can view and query CloudTrail activity on an account through CloudWatch Logs and CloudWatch Logs Insights. This activity includes AWS Control Tower lifecycle events. CloudWatch Logs’ capabilities allow you to perform more granular and precise queries than you would normally be able to make using CloudTrail.

For more information, see Logging AWS Control Tower Actions with AWS CloudTrail.

Logging AWS Control Tower Actions with AWS CloudTrail

AWS Control Tower is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Control Tower. CloudTrail captures actions for AWS Control Tower as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Control Tower.

If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Control Tower, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

AWS Control Tower Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS Control Tower, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

> **Note**
>
> In AWS Control Tower releases before landing zone version 3.0, AWS Control Tower created a member account trail. When you update to release 3.0, your CloudTrail trail is updated to become an organization trail. For best practices when moving between trails, see Creating an organizational trail in the CloudTrail User Guide.

Recommended: Create a trail

For an ongoing record of events in your AWS account, including events for AWS Control Tower, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- Prepare for creating a trail
- Managing CloudTrail costs
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

AWS Control Tower logs the following actions as events in CloudTrail log files:

Public APIs

- For a full list of the AWS Control Tower public APIs and details about each one, see The AWS Control Tower API Reference. Calls to these public APIs are logged by AWS CloudTrail.

Other APIs

- `SetupLandingZone`
- `UpdateAccountFactoryConfig`
- `ManageOrganizationalUnit`
- `CreateManagedAccount`
- `GetLandingZoneStatus`
- `GetHomeRegion`
- `ListManagedAccounts`
- `DescribeManagedAccount`
- `DescribeAccountFactoryConfig`
- `DescribeGuardrailForTarget`
- `DescribeManagedOrganizationalUnit`
- `ListEnabledGuardrails`
- `ListGuardrailViolations`
- `ListGuardrails`
- `ListGuardrailsForTarget`
- `ListManagedAccountsForGuardrail`

- `ListManagedAccountsForParent`
- `ListManagedOrganizationalUnits`
- `ListManagedOrganizationalUnitsForGuardrail`
- `GetGuardrailComplianceStatus`
- `DescribeGuardrail`
- `ListDirectoryGroups`
- `DescribeSingleSignOn`
- `DescribeCoreService`
- `GetAvailableUpdates`

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.
- Whether the request was rejected as access denied or processed successfully.

For more information, see the CloudTrail userIdentity Element.

Example: AWS Control Tower Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail events don't appear in any specific order in the log files.

The following example shows a CloudTrail log entry that shows the structure of a typical log file entry for a `SetupLandingZone` AWS Control Tower event, including a record of the identity of the user who initiated the action.

```
{
  "eventVersion": "1.05",
```

```
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:backend-test-assume-role-session",
    "arn": "arn:aws:sts::76543EXAMPLE;:assumed-role/AWSControlTowerTestAdmin/backend-
test-assume-role-session",
    "accountId": "76543EXAMPLE",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-11-20T19:36:11Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::AKIAIOSFODNN7EXAMPLE:role/AWSControlTowerTestAdmin",
        "accountId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "AWSControlTowerTestAdmin"
      }
    }
  },
  "eventTime": "2018-11-20T19:36:15Z",
  "eventSource": "controltower.amazonaws.com",
  "eventName": "SetupLandingZone",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "Coral/Netty4",
  "errorCode": "InvalidParametersException",
  "errorMessage": "Home region EU_CENTRAL_1 is unsupported",
  "requestParameters": {
    "homeRegion": "EU_CENTRAL_1",
    "logAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "sharedServiceAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "securityAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "securityNotificationEmail": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "responseElements": null,
  "requestID": "96f47b68-ed5f-4268-931c-807cd1f89a96",
  "eventID": "4ef5cf08-39e5-4fdf-9ea2-b07ced506851",
  "eventType": "AwsApiCall",
  "recipientAccountId": "76543EXAMPLE"
}
```

Monitor resource changes with AWS Config

AWS Control Tower enables AWS Config on all enrolled accounts, so that it can monitor compliance through detective controls, record resource changes, and deliver resource change logs to the log archive account.

If your landing zone version is earlier than 3.0: For your enrolled accounts, AWS Config logs all changes to resources, for all Regions in which the account operates. Each change is modeled as a configuration item (CI), which contains information such as the resource identifier, the Region, the date that each change was recorded, and whether the change relates to a known resource or a newly discovered one.

If your landing zone version is 3.0 or later: AWS Control Tower limits recording for global resources, such as IAM users, groups, roles, and customer managed polices, to your home Region only. Copies of global resource changes are not stored in every Region. This limitation of resource recording conforms with AWS Config best practices. A full list of global resources is available in AWS Config documentation.

- To learn more about AWS Config, see How AWS Config works.
- For a list of resources that AWS Config can support, see Supported resource types.
- To learn about how to customize resource tracking in the AWS Control Tower environment, see the blog post entitled Customize AWS Config resource tracking in AWS Control Tower.

AWS Control Tower sets up an AWS Config delivery channel in all enrolled accounts. Through this delivery channel, it logs all changes recorded by AWS Config in the log archive account, where they are stored to a folder in an Amazon Simple Storage Service bucket.

Manage AWS Config costs in AWS Control Tower

This section describes how AWS Config records and bills you for changes to resources in your AWS Control Tower accounts. This information may help you understand how to manage the costs associated with AWS Config, when you're utilizing AWS Control Tower. AWS Control Tower adds no additional cost.

> **Note**
>
> **If your landing zone version is 3.0 or later**: AWS Control Tower limits AWS Config recording for global resources, such as IAM users, groups, roles, and customer-managed

polices, to your home Region only. Therefore, some of the information in this section may not apply to your landing zone.

AWS Config is designed to record each change to each resource, in each Region where an account operates, as a configuration item (CI). AWS Config bills you for each configuration item that it generates.

How AWS Config operates

AWS Config records resources in each Region, separately. Some global resources, such as IAM roles, are recorded once per Region. For example, if you create a new IAM role in an enrolled account that is operating in five Regions, AWS Config generates five CIs, one for each Region. Other global resources, such as Route 53 hosted zones, are recorded only once across all Regions. For example, if you create a new Route 53 hosted zone in an enrolled account, AWS Config generates one CI, regardless of how many Regions are selected for that account. For a list that helps you distinguish these types of resources, see [The same resource is recorded multiple times](#).

> **Note**
>
> When AWS Control Tower works with AWS Config, a Region may be governed by AWS Control Tower, or ungoverned, and AWS Config still records the changes if the account operates in that Region.

AWS Config detects two types of relationships in resources

AWS Config makes a distinction between *direct* and *indirect* relationships among resources. If a resource is returned in another resource's **Describe** API call, those resources are recorded as a direct relationship. When you change a resource in a direct relationship with another resource, AWS Config does not make a CI for both resources.

For example, if you create an Amazon EC2 instance, and the API requires you to create a network interface, AWS Config considers the Amazon EC2 instance to have a direct relationship with the network interface. As a result, AWS Config generates only one CI.

AWS Config records separate changes for resource relationships that are *indirect* relationships. For example, AWS Config generates two CIs if you create a security group and add an associated Amazon EC2 instance that's part of the security group.

For more information about direct and indirect relationships, see What is a direct and an indirect relationship with respect to a resource?

You can find a list of resource relationships in the AWS Config documentation.

View the AWS Config recorder data on enrolled accounts

AWS Config is integrated with CloudWatch so that you can view AWS Config CIs in a dashboard. For more information, see the blog post entitled AWS Config supports Amazon CloudWatch metrics.

Programmatically, to view AWS Config data, you can work with the AWS CLI, or you can utilize other AWS tools.

Query the AWS Config recorder data on a specific resource

You can use the AWS CLI to retrieve a list of the most recent changes for a resource.

Resource history command:

- `aws configservice get-resource-config-history --resource-type RESOURCE-TYPE --resource-id RESOURCE-ID --region REGION`

To learn more, see the API documentation for `get-config-history`.

Visualize AWS Config data with QuickSight

You can visualize and query resources recorded by AWS Config across your entire organization. For more information, see the Config Resource Compliance Dashboard and Visualizing AWS Config data using Amazon Athena and QuickSight.

Troubleshooting AWS Config in AWS Control Tower

This section gives information about some problems you may encounter when using AWS Config with AWS Control Tower.

High AWS Config costs

If your workflow includes processes that create, update, or delete resources frequently, or if it handles resources in large numbers, that workflow may generate large numbers of CIs. If you run these processes in a non-production account, consider unenrolling the account. You may need to de-activate the AWS Config recorder for that account manually.

> **Note**
>
> After you unenroll the account, AWS Control Tower cannot enforce detective controls or log account events, such as AWS Config activities, for resources in that account.

For more information, see Unmanage an enrolled account. To learn how to deactivate the AWS Config recorder, see Managing the configuration recorder.

The same resource is recorded multiple times

Check whether the resource is a global resource. For AWS Control Tower landing zones prior to version 3.0, AWS Config may record certain global resources once for each Region in which AWS Config is operating. For example, if AWS Config is enabled on eight Regions, each role is recorded eight times.

The following resources are recorded once for each Region in which AWS Config is operating:

- `AWS::IAM::Group`
- `AWS::IAM::Policy`
- `AWS::IAM::Role`
- `AWS::IAM::User`

Other global resources are recorded only once. Here are some examples of resources that are recorded once:

- `AWS::Route53::HostedZone`
- `AWS::Route53::HealthCheck`
- `AWS::ECR::PublicRepository`
- `AWS::GlobalAccelerator::Listener`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::GlobalAccelerator::Accelerator`

AWS Config did not record a resource

Certain resources have dependency relationships with other resources. These relationships may be *direct* or *indirect*. You can find a list of deprecated indirect relationships in the AWS Config FAQ.

Lifecycle Events in AWS Control Tower

Some events logged by AWS Control Tower are *lifecycle events*. A lifecycle event's purpose is to mark the *completion* of certain AWS Control Tower actions that change the state of resources. Lifecycle events apply to resources that AWS Control Tower creates or manages, such as a landing zone, baseline, or control, related to an organizational unit (OU) or account.

Characteristics of AWS Control Tower lifecycle events

- For each lifecycle event, the event log shows whether the originating Control Tower action completed successfully, or failed.
- AWS CloudTrail automatically records each lifecycle event as a *non-API AWS service event*. For more information, see the AWS CloudTrail User Guide.
- Each lifecycle event also is delivered to the Amazon EventBridge and Amazon CloudWatch Events services.

Lifecycle events in AWS Control Tower offer two primary benefits:

- Because a lifecycle event registers the completion of an AWS Control Tower action, you can create an Amazon EventBridge rule or Amazon CloudWatch Events rule that can trigger the next steps in your automation workflow, based on the state of the lifecycle event.
- The logs provide additional detail to assist administrators and auditors in reviewing certain types of activity in your organizations.

How lifecycle events work

AWS Control Tower relies upon multiple services to implement its actions. Therefore, each lifecycle event is recorded only after a series of actions is complete. For example, when you enable a control on an OU, AWS Control Tower launches a series of sub-steps that implement the request. The final result of the entire series of sub-steps is recorded in the log as the state of the lifecycle event.

- If every underlying sub-step has completed successfully, the lifecycle event state is recorded as **Succeeded**.

- If any of the underlying sub-steps did not complete successfully, the lifecycle event state is recorded as **Failed**.

Each lifecycle event includes a logged timestamp that shows when the AWS Control Tower action was initiated, and another timestamp showing when the lifecycle event is completed, marking success or failure.

Viewing lifecycle events in Control Tower

You can view lifecycle events from the **Activities** page in your AWS Control Tower dashboard.

- To navigate to the **Activities** page, choose **Activities** from the left navigation pane.
- To get more details about a specific event, select the event and then choose the **View details** button at the upper right.

For more information about how to integrate AWS Control Tower lifecycle events into your workflows, see this blog post, [Using lifecycle events to track AWS Control Tower actions and trigger automated workflows](#).

Expected behavior of CreateManagedAccount and UpdateManagedAccount lifecycle events

When you create an account or enroll an account in AWS Control Tower, those two actions call the same internal API. If there's an error during the process, it usually occurs after the account has been created but is not fully provisioned. When you retry to create the account after the error, or when you try to update the provisioned product, AWS Control Tower sees that the account already exists.

Because the account exists, AWS Control Tower records the `UpdateManagedAccount` lifecycle event instead of the `CreateManagedAccount` lifecycle event at the end of the retry request. You may have expected to see another `CreateManagedAccount` event because of the error. However, the `UpdateManagedAccount` lifecycle event is the expected and desired behavior.

If you plan to create or enroll accounts into AWS Control Tower using automated methods, program the Lambda function to look for **UpdateManagedAccount** lifecycle events as well as **CreateManagedAccount** lifecycle events.

Lifecycle event names

Each lifecycle event is named so that it corresponds to the originating AWS Control Tower action, which also is recorded by AWS CloudTrail. Thus, for example, a lifecycle event originated by the AWS Control Tower `CreateManagedAccount` CloudTrail event is named `CreateManagedAccount`.

Each name in the list that follows is a link to an example of the logged detail in `JSON` format. The additional detail shown in these examples is taken from the Amazon CloudWatch event logs.

Although `JSON` does not support comments, some comments have been added in the examples for explanatory purposes. Comments are preceded by "//" and they appear in the right side of the examples.

In these examples, some account names and organization names are obscured. An `accountId` is always a 12-number sequence, which has been replaced with "xxxxxxxxxxxx" in the examples. An `organizationalUnitID` is a unique string of letters and numbers. Its form is preserved in the examples.

- `CreateManagedAccount`: The log records whether AWS Control Tower successfully completed every action to create and provision a new account using account factory.
- `UpdateManagedAccount`: The log records whether AWS Control Tower successfully completed every action to update a provisioned product that's associated with an account you had previously created by using account factory.
- `EnableGuardrail`: The log records whether AWS Control Tower successfully completed every action to enable a control on an OU that was created by AWS Control Tower.
- `DisableGuardrail`: The log records whether AWS Control Tower successfully completed every action to disable a control on an OU that was created by AWS Control Tower.
- `SetupLandingZone`: The log records whether AWS Control Tower successfully completed every action to set up a landing zone.
- `UpdateLandingZone`: The log records whether AWS Control Tower successfully completed every action to update your existing landing zone.
- `RegisterOrganizationalUnit`: The log records whether AWS Control Tower successfully completed every action to enable its governance features on an OU.
- `DeregisterOrganizationalUnit`: The log records whether AWS Control Tower successfully completed every action to disable its governance features on an OU.
- `PrecheckOrganizationalUnit`: The log records whether AWS Control Tower detected any resource that would would prevent the **Extend governance** operation from completing successfully.

- EnableBaseline: The log records whether AWS Control Tower successfully completed every action to enable a new baseline on a target member account under an OU. The enable operation can be initiated using the `EnableBaseline` API or the console.
- ResetEnabledBaseline: The log records whether AWS Control Tower successfully completed every action to reset an existing enabled baseline on a target member account under an OU. The reset operation can be initiated using the `ResetEnabledBaseline` API or the console.
- UpdateEnabledBaseline: The log records whether AWS Control Tower successfully completed every action to update an existing enabled baseline on a target member account under an OU. The update operation can be initiated using the `UpdateEnabledBaseline` API or the console.
- DisableBaseline: The log records whether AWS Control Tower successfully completed every action to disable an existing enabled baseline on a target member account under an OU. The disable operation can be initiated using the `DisableBaseline` API or the console.

The following sections provide a list of AWS Control Tower lifecycle events, with examples of the details logged for each type of lifecycle event.

CreateManagedAccount

This lifecycle event records whether AWS Control Tower successfully created and provisioned a new account using account factory. This event corresponds to the AWS Control Tower `CreateManagedAccount` CloudTrail event. The lifecycle event log includes the `accountName` and `accountId` of the newly-created account, and the `organizationalUnitName` and `organizationalUnitId` of the OU in which the account has been placed.

```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",                                   // Management account
 ID.
    "time": "2018-08-30T21:42:18Z",                              // Format: yyyy-MM-
dd'T'hh:mm:ssZ
    "region": "us-east-1",                                       // AWS Control Tower
 home region.
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
```

```
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",                     // Timestamp when call
 was made. Format: yyyy-MM-dd'T'hh:mm:ssZ.
        "eventSource": "controltower.amazonaws.com",
        "eventName": "CreateManagedAccount",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "createManagedAccountStatus": {
                "organizationalUnit":{
                    "organizationalUnitName":"Custom",
                    "organizationalUnitId":"ou-XXXX-l3zc8b3h"

                    },
                "account":{
                    "accountName":"LifeCycle1",
                    "accountId":"XXXXXXXXXXXX"
                    },
                "state":"SUCCEEDED",
                "message":"AWS Control Tower successfully created a managed account.",
                "requestedTimestamp":"2019-11-15T11:45:18+0000",
                "completedTimestamp":"2019-11-16T12:09:32+0000"}
        }
    }
}
```

UpdateManagedAccount

This lifecycle event records whether AWS Control Tower successfully updated the provisioned product associated with an account that was created previously by using account factory. This event corresponds to the AWS Control Tower `UpdateManagedAccount` CloudTrail event. The lifecycle event log includes the `accountName` and `accountId` of the associated account, and the `organizationalUnitName` and `organizationalUnitId` of the OU in which the updated account is placed.

```
{
```

```
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",                                  // AWS Control Tower
 organization management account.
    "time": "2018-08-30T21:42:18Z",                             // Format: yyyy-MM-
dd'T'hh:mm:ssZ
    "region": "us-east-1",                                      // AWS Control Tower
 home region.
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",                    // Timestamp when call
 was made. Format: yyyy-MM-dd'T'hh:mm:ssZ.
        "eventSource": "controltower.amazonaws.com",
        "eventName": "UpdateManagedAccount",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "updateManagedAccountStatus": {
                "organizationalUnit":{
                    "organizationalUnitName":"Custom",
                    "organizationalUnitId":"ou-XXXX-l3zc8b3h"
                    },
                "account":{
                    "accountName":"LifeCycle1",
                    "accountId":"XXXXXXXXXXXX"
                    },
                "state":"SUCCEEDED",
                "message":"AWS Control Tower successfully updated a managed account.",
                "requestedTimestamp":"2019-11-15T11:45:18+0000",
                "completedTimestamp":"2019-11-16T12:09:32+0000"}
        }
    }
```

```
}
```

EnableGuardrail

This lifecycle event records whether AWS Control Tower successfully enabled a control on an OU that is being managed by AWS Control Tower. This event corresponds to the AWS Control Tower `EnableGuardrail` CloudTrail event. The lifecycle event log includes the `guardrailId` and `guardrailBehavior` of the control, and the `organizationalUnitName` and `organizationalUnitId` of the OU on which the control is enabled.


```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",
    "time": "2018-08-30T21:42:18Z",                              // End-time of action.
 Format: yyyy-MM-dd'T'hh:mm:ssZ
    "region": "us-east-1",                                       // AWS Control Tower
 home region.
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",
        "eventSource": "controltower.amazonaws.com",
        "eventName": "EnableGuardrail",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "enableGuardrailStatus": {
                "organizationalUnits": [
                    {
                      "organizationalUnitName": "Custom",
                      "organizationalUnitId": "ou-vwxy-18vy4yro"
```

```
                    }
                  ],
                  "guardrails": [
                    {
                      "guardrailId": "AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK",
                      "guardrailBehavior": "DETECTIVE"
                    }
                  ],
                  "state": "SUCCEEDED",
                  "message": "AWS Control Tower successfully enabled a guardrail on an
 organizational unit.",
                  "requestTimestamp": "2019-11-12T09:01:07+0000",
                  "completedTimestamp": "2019-11-12T09:01:54+0000"
                }
        }
    }
}
```


DisableGuardrail

This lifecycle event records whether AWS Control Tower successfully disabled a control on an OU that is being managed by AWS Control Tower. This event corresponds to the AWS Control Tower `DisableGuardrail` CloudTrail event. The lifecycle event log includes the `guardrailId` and `guardrailBehavior` of the control, and the `organizationalUnitName` and `organizationalUnitId` of the OU on which the control is disabled.


```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",
    "time": "2018-08-30T21:42:18Z",
    "region": "us-east-1",
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",
```

```
        "eventSource": "controltower.amazonaws.com",
        "eventName": "DisableGuardrail",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "disableGuardrailStatus": {
                   "organizationalUnits": [
                    {
                      "organizationalUnitName": "Custom",
                      "organizationalUnitId": "ou-vwxy-18vy4yro"
                    }
                  ],
                  "guardrails": [
                    {
                      "guardrailId": "AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK",
                      "guardrailBehavior": "DETECTIVE"
                    }
                  ],
                  "state": "SUCCEEDED",
                  "message": "AWS Control Tower successfully disabled a guardrail on an
 organizational unit.",
                  "requestTimestamp": "2019-11-12T09:01:07+0000",
                  "completedTimestamp": "2019-11-12T09:01:54+0000"
                }
            }
        }
    }
```


SetupLandingZone

This lifecycle event records whether AWS Control Tower successfully set up a landing zone. This event corresponds to the AWS Control Tower `SetupLandingZone` CloudTrail event. The lifecycle event log includes the `rootOrganizationalId`, which is ID of the organization that AWS Control Tower creates from the management account. The log entry also includes the `organizationalUnitName` and `organizationalUnitId` for each of of the OUs, and the `accountName` and `accountId` for each account, that are created when AWS Control Tower sets up the landing zone.

```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",                // Request ID.
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",                                   // Management account
 ID.
    "time": "2018-08-30T21:42:18Z",                              // Event time from
 CloudTrail.
    "region": "us-east-1",                                       // Management account
 CloudTrail region.
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",                         // Management-account
 ID.
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",                     // Timestamp when call
 was made. Format: yyyy-MM-dd'T'hh:mm:ssZ.
        "eventSource": "controltower.amazonaws.com",
        "eventName": "SetupLandingZone",
        "awsRegion": "us-east-1",                                // AWS Control Tower
 home region.
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "CloudTrail_event_ID",                        // This value is
 generated by CloudTrail.
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "setupLandingZoneStatus": {
                "state": "SUCCEEDED",                             // Status of entire
 lifecycle operation.
                "message": "AWS Control Tower successfully set up a new landing zone.",

                "rootOrganizationalId" : "r-1234",
                "organizationalUnits" : [                         // Use a list.
                  {
                    "organizationalUnitName": "Security",             // Security OU
 name.
                    "organizationalUnitId": "ou-adpf-302pk332"    // Security OU ID.
```

```
                    },
                    {
                      "organizationalUnitName": "Custom",           // Custom OU name.
                      "organizationalUnitId": "ou-adpf-302pk332"    // Custom OU ID.
                    },
                  ],
                 "accounts": [                                      // All created
 accounts are here. Use a list of "account" objects.

                    {
                      "accountName": "Audit",
                      "accountId": "XXXXXXXXXXXX"
                    },
                    {
                      "accountName": "Log archive",
                      "accountId": "XXXXXXXXXXXX"
                    }
                ],
                "requestedTimestamp": "2018-08-30T21:42:18Z",
                "completedTimestamp": "2018-08-30T21:42:18Z"
            }
        }
    }
}
```

UpdateLandingZone

This lifecycle event records whether AWS Control Tower successfully updated your existing landing zone. This event corresponds to the AWS Control Tower `UpdateLandingZone` CloudTrail event. The lifecycle event log includes the `rootOrganizationalId`, which is ID of the (updated) organization governed by AWS Control Tower. The log entry also includes the `organizationalUnitName` and `organizationalUnitId` for each of of the OUs, and the `accountName` and `accountId` for each account, that was created previously, when AWS Control Tower originally set up the landing zone.

```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",                // Request ID.
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",                                   // Management account
 ID.
```

```
    "time": "2018-08-30T21:42:18Z",                                // Event time from
CloudTrail.
    "region": "us-east-1",                                         // Management account
CloudTrail region.
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",                           // Management account
ID.
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",                       // Timestamp when call
was made. Format: yyyy-MM-dd'T'hh:mm:ssZ.
        "eventSource": "controltower.amazonaws.com",
        "eventName": "UpdateLandingZone",
        "awsRegion": "us-east-1",                                  // AWS Control Tower
home region.
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "CloudTrail_event_ID",                          // This value is
generated by CloudTrail.

        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "updateLandingZoneStatus": {
                "state": "SUCCEEDED",                              // Status of entire
operation.
                "message": "AWS Control Tower successfully updated a landing zone.",


                "rootOrganizationalId" : "r-1234",
                "organizationalUnits" : [                           // Use a list.
                  {
                    "organizationalUnitName": "Security",             // Security OU
name.
                    "organizationalUnitId": "ou-adpf-302pk332"    // Security OU ID.
                  },
                  {
                    "organizationalUnitName": "Custom",             // Custom OU name.
                    "organizationalUnitId": "ou-adpf-302pk332"     // Custom OU ID.
                  },
                ],
```

```
                "accounts": [                                         // All created
 accounts are here. Use a list of "account" objects.

                  {
                    "accountName": "Audit",
                    "accountId": "XXXXXXXXXXXX"
                  },
                  {
                    "accountName": "Log archive",
                    "accountId": "XXXXXXXXXX"
                  }
              ],
              "requestedTimestamp": "2018-08-30T21:42:18Z",
              "completedTimestamp": "2018-08-30T21:42:18Z"
            }
        }
    }
}
```


RegisterOrganizationalUnit

This lifecycle event records whether AWS Control Tower successfully enabled its governance features on an OU. This event corresponds to the AWS Control Tower `RegisterOrganizationalUnit` CloudTrail event. The lifecycle event log includes the `organizationalUnitName` and `organizationalUnitId` of the OU that AWS Control Tower has brought under its governance.


```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "123456789012",
    "time": "2018-08-30T21:42:18Z",
    "region": "us-east-1",
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
```

```
        "eventTime": "2018-08-30T21:42:18Z",
        "eventSource": "controltower.amazonaws.com",
        "eventName": "RegisterOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "registerOrganizationalUnitStatus": {
                "state": "SUCCEEDED",

                "message": "AWS Control Tower successfully registered an organizational
 unit.",

                "organizationalUnit" :
                  {
                    "organizationalUnitName": "Test",
                    "organizationalUnitId": "ou-adpf-302pk332"
                  }
                "requestedTimestamp": "2018-08-30T21:42:18Z",
                "completedTimestamp": "2018-08-30T21:42:18Z"
            }
        }
    }
}
```


DeregisterOrganizationalUnit

This lifecycle event records whether AWS Control Tower successfully disabled its governance features on an OU. This event corresponds to the AWS Control Tower `DeregisterOrganizationalUnit` CloudTrail event. The lifecycle event log includes the `organizationalUnitName` and `organizationalUnitId` of the OU on which AWS Control Tower has disabled its governance features.


```
{
    "version": "0",
    "id": "999cccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",
```

```
    "time": "2018-08-30T21:42:18Z",
    "region": "us-east-1",
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",
        "eventSource": "controltower.amazonaws.com",
        "eventName": "DeregisterOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "deregisterOrganizationalUnitStatus": {
                "state": "SUCCEEDED",
                "message": "AWS Control Tower successfully deregistered an
  organizational unit, and enabled mandatory guardrails on the new organizational
  unit.",
                "organizationalUnit" :
                  {
                    "organizationalUnitName": "Test",                  // Foundational
  OU name.
                    "organizationalUnitId": "ou-adpf-302pk332"         // Foundational
  OU ID.
                  },
                "requestedTimestamp": "2018-08-30T21:42:18Z",
                "completedTimestamp": "2018-08-30T21:42:18Z"
            }
            }
        }
    }
```


PrecheckOrganizationalUnit

This lifecycle event records whether AWS Control Tower successfully performed prechecks on an OU. This event corresponds to the AWS Control Tower `PrecheckOrganizationalUnit` CloudTrail event. The lifecycle event log contains a field for the `Id`, `Name`, and `failedPrechecks` values, for each resource on which AWS Control Tower has performed prechecks during the OU registration process.

The event log also contains information about the nested accounts on which the prechecks were performed, including the `accountName`, `accountId`, and `failedPrechecks` fields.

If the `failedPrechecks` value is empty, it means that all prechecks for that resource passed successfully.

- This event is emitted only if there is a precheck failure.
- This event is not emitted if you are registering an empty OU.

Example of event:


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "XXXXXXXXXXXX",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2021-09-20T22:45:43Z",
  "eventSource": "controltower.amazonaws.com",
  "eventName": "PrecheckOrganizationalUnit",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "eventID": "b41a9d67-0da4-4dc5-a87a-25fa19dc5305",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "XXXXXXXXXXXX",
  "serviceEventDetails": {
    "precheckOrganizationalUnitStatus": {
      "organizationalUnit": {
        "organizationalUnitName": "Ou-123",
        "organizationalUnitId": "ou-abcd-123456",
        "failedPrechecks": [
            "SCP_CONFLICT"
          ]
      },
      "accounts": [
        {
```

```
          "accountName": "Child Account 1",
          "accountId": "XXXXXXXXXXXX",
          "failedPrechecks": [
            "FAILED_TO_ASSUME_ROLE"
          ]
        },
        {
          "accountName": "Child Account 2",
          "accountId": "XXXXXXXXXXXX",
          "failedPrechecks": [
            "FAILED_TO_ASSUME_ROLE"
          ]
        },
        {
          "accountName": "Management Account",
          "accountId": "XXXXXXXXXXXX",
          "failedPrechecks": [
            "MISSING_PERMISSIONS_AF_PRODUCT"
          ]
        },
        {
          "accountName": "Child Account 3",
          "accountId": "XXXXXXXXXXXX",
          "failedPrechecks": []
        },
        ...
      ],
      "state": "FAILED",
      "message": "AWS Control Tower failed to register an organizational unit due to
 pre-check failures. Go to the OU details page to download a list of failed pre-checks
 for the OU and accounts within.",
      "requestedTimestamp": "2021-09-20T22:44:02+0000",
      "completedTimestamp": "2021-09-20T22:45:43+0000"
    }
  },
  "eventCategory": "Management"
}
```


EnableBaseline

This lifecycle event records whether AWS Control Tower successfully enabled a baseline on a target member account under an OU. This event corresponds to the AWS Control Tower `RegisterOrganizationalUnit` or `EnableBaseline` CloudTrail events. The lifecycle event log includes the baseline that was enabled and its version, the `targetIdentifier` on which the baseline was enabled, the `parentIdentifier` of the baseline enabled on the parent OU, and the `statusSummary` showing the SUCCEEDED or FAILED status, along with the additional parameters and timestamp of the operation.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "XXXXXXXXXXXX",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-02-10T17:14:57Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "EnableBaseline",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "366911a2-4fa6-4e4a-ac2b-280f627e0027",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "XXXXXXXXXXXX",
    "serviceEventDetails": {
        "enableBaselineStatus": {
            "enabledBaselineDetails": {
                "arn": "arn:aws:controltower:us-east-2:XXXXXXXXXXXX:enabledbaseline/
XXXXXXXXXXXXXXXX",
                "parentIdentifier": "arn:aws:controltower:us-
east-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX",
                "targetIdentifier": "arn:aws:organizations::XXXXXXXXXXXX:account/o-
ern76xmzvf/XXXXXXXXXXXX",
                "baselineIdentifier": "arn:aws:controltower:us-east-2::baseline/
XXXXXXXXXXXXXXXX",
                "baselineVersion": "4.0",
                "statusSummary": {
                    "lastOperationIdentifier": "37f5eb68-e5b9-4c70-ae76-4ca15f6b16de",
                    "status": "SUCCEEDED"
                },
                "parameters": [
                    {
                        "key": "IdentityCenterEnabledBaselineArn",
```

```
                        "value": {
                            "untyped": {
                                "object": "arn:aws:controltower:us-
east-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX"                            }
                        }
                    }
                ]
            },
            "requestedTimestamp": "2025-02-10T17:07:09+0000",
            "completedTimestamp": "2025-02-10T17:14:57+0000"
        }
    },
    "eventCategory": "Management"
}
```


ResetEnabledBaseline

This lifecycle event records whether AWS Control Tower successfully reset the existing enabled baseline on a target member account under an OU. This event corresponds to the AWS Control Tower `RegisterOrganizationalUnit` or `ResetEnabledBaseline` CloudTrail events. The lifecycle event log includes the baseline that was enabled and its version, the `targetIdentifier` on which the baseline was enabled, the `parentIdentifier` of the baseline enabled on the parent OU, and the `statusSummary` showing the SUCCEEDED or FAILED status, along with the additional parameters and timestamp of the operation.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "XXXXXXXXXXXX",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-02-10T21:17:55Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "ResetEnabledBaseline",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "c01a32e1-13ab-4b46-8f1b-00699ef6f989",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
```

```
    "managementEvent": true,
    "recipientAccountId": "XXXXXXXXXXXX",
    "serviceEventDetails": {
        "resetEnabledBaselineStatus": {
            "enabledBaselineDetails": {
                "arn": "arn:aws:controltower:us-west-2:XXXXXXXXXXXX:enabledbaseline/
XXXXXXXXXXXXXXXX",
                "parentIdentifier": "arn:aws:controltower:us-
west-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX",
                "targetIdentifier": "arn:aws:organizations::XXXXXXXXXXXX:account/
o-0uh2kplf6d/XXXXXXXXXXXX",
                "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline/
XXXXXXXXXXXXXXX",
                "baselineVersion": "1.0",
                "statusSummary": {
                    "lastOperationIdentifier": "3e364c89-89fa-42b8-9776-9f7cc47ba1fa",
                    "status": "SUCCEEDED"
                },
                "parameters": []
            },
            "requestedTimestamp": "2025-02-10T21:14:24Z",
            "completedTimestamp": "2025-02-10T21:17:54+0000"
        }
    },
    "eventCategory": "Management"
}
```


UpdateEnabledBaseline

This lifecycle event records whether AWS Control Tower successfully updated the existing enabled baseline on a target member account under an OU. This event corresponds to the AWS Control Tower `RegisterOrganizationalUnit` or `UpdateEnabledBaseline` CloudTrail events. The lifecycle event log includes the baseline that was enabled and its version, the `targetIdentifier` on which the baseline was enabled, the `parentIdentifier` of the baseline enabled on the parent OU, and the `statusSummary` showing the SUCCEEDED or FAILED status, along with the additional parameters and timestamp of the operation.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "XXXXXXXXXXXX",
        "invokedBy": "AWS Internal"
```

```
    },
    "eventTime": "2025-02-10T19:45:28Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "UpdateEnabledBaseline",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "514f2aff-1a99-4912-bda1-0d4d6662c96e",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "XXXXXXXXXXXX",
    "serviceEventDetails": {
        "updateEnabledBaselineStatus": {
            "enabledBaselineDetails": {
                "arn": "arn:aws:controltower:us-east-2:XXXXXXXXXXXX:enabledbaseline/
XXXXXXXXXXXXXXXX",
                "parentIdentifier": "arn:aws:controltower:us-
east-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX",
                "targetIdentifier": "arn:aws:organizations::XXXXXXXXXXXX:account/o-
ern76xmzvf/XXXXXXXXXXXX",
                "baselineIdentifier": "arn:aws:controltower:us-east-2::baseline/
XXXXXXXXXXXXXXX",
                "baselineVersion": "4.0",
                "statusSummary": {
                    "lastOperationIdentifier": "ba3de28f-83fb-4c9a-8a8c-a4e15fac2c41",
                    "status": "SUCCEEDED"
                },
                "parameters": [
                    {
                        "key": "IdentityCenterEnabledBaselineArn",
                        "value": {
                            "untyped": {
                                "object": "arn:aws:controltower:us-
east-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX"                            }
                        }
                    }
                ]
            },
            "requestedTimestamp": "2025-02-10T19:39:35+0000",
            "completedTimestamp": "2025-02-10T19:45:28+0000"
        }
```

```
    },
    "eventCategory": "Management"
}
```

DisableBaseline

This lifecycle event records whether AWS Control Tower successfully disabled the existing enabled baseline on a target member account under an OU. This event corresponds to the AWS Control Tower `DisableBaseline` CloudTrail event. The lifecycle event log includes the baseline that was enabled and its version, the `targetIdentifier` on which the baseline was enabled, the `parentIdentifier` of the baseline enabled on the parent OU, and the `statusSummary` showing the SUCCEEDED or FAILED status, along with the additional parameters and timestamp of the operation.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "XXXXXXXXXXXX",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-03-14T00:50:58Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "DisableBaseline",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "704794c4-a32e-4960-8386-c7efaa5a22a1",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "XXXXXXXXXXXX",
    "serviceEventDetails": {
        "disableBaselineStatus": {
            "enabledBaselineDetails": {
                "arn": "arn:aws:controltower:us-west-2:XXXXXXXXXXXX:enabledbaseline/
XXXXXXXXXXXXXXXX",
                "parentIdentifier": "arn:aws:controltower:us-
west-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX",
                "targetIdentifier": "arn:aws:organizations::XXXXXXXXXXXX:account/
o-0uh2kplf6d/XXXXXXXXXXXX",
```

```
                "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline/
XXXXXXXXXXXXXXX",
                "baselineVersion": "1.0",
                "statusSummary": {
                    "lastOperationIdentifier": "7b895594-0edb-48bc-9f3d-d88c2ad618df",
                    "status": "SUCCEEDED"
                },
                "parameters": []
            },
            "baselineDetails": {
                "arn": "arn:aws:controltower:us-west-2:XXXXXXXXXXXX:enabledbaseline/
XXXXXXXXXXXXXXXX",
                "parentIdentifier": "arn:aws:controltower:us-
west-2:XXXXXXXXXXXX:enabledbaseline/XXXXXXXXXXXXXXXX",
                "targetIdentifier": "arn:aws:organizations::XXXXXXXXXXXX:account/
o-0uh2kplf6d/XXXXXXXXXXXX",
                "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline/
XXXXXXXXXXXXXXX",
                "baselineVersion": "1.0",
                "statusSummary": {
                    "lastOperationIdentifier": "7b895594-0edb-48bc-9f3d-d88c2ad618df",
                    "status": "SUCCEEDED"
                },
                "parameters": []
            },
            "requestedTimestamp": "2025-03-14T00:49:13Z",
            "completedTimestamp": "2025-03-14T00:50:58+0000"
        }
    },
    "eventCategory": "Management"
}
```


Using AWS User Notifications with AWS Control Tower

You can use AWS User Notifications to set up delivery channels to be notified about AWS Control Tower events. You receive a notification when an event matches a rule that you specify. You can receive notifications for events through multiple channels, including email, Amazon Q Developer in chat applications chat notifications, or AWS Console Mobile App push notifications. You can also see notifications in the Console Notifications Center.

AWS User Notifications supports aggregation, which can reduce the number of notifications you receive during specific events. Notifications also are visible in the Console Notifications Center.

The advantages of subscribing to notifications through AWS User Notifications instead of EventBridge include:

- A friendlier user interface (UI).
- Integration with the AWS console, in the bell/notifications area on the global navigation bar.
- Native support for email notifications, there's no need to set up Amazon SNS.
- Most notably, support for mobile push notifications, exclusive to AWS User Notifications.

For example, one type of notification you may wish to receive is in case of Security Hub critical and high severity findings. A code snippet in JSON to set up that notification subscription may look something like this:

```
{
  "detail": {
    "findings": {
      "Compliance": {
        "Status": ["FAILED", "WARNING", "NOT_AVAILABLE"]
      },
      "RecordState": ["ACTIVE"],
      "Severity": {
        "Label": ["CRITICAL", "HIGH"]
      },
      "Workflow": {
        "Status": ["NEW", "NOTIFIED"]
      }
    }
  }
}
```

Event filtering

- You can filter events by service and name using the filters available on the AWS User Notifications console.
- You can filter events by specific properties if you create your own EventBridge filter from JSON code.

Example AWS Control Tower event

Here is a generalized example event for AWS Control Tower.

- It an EventBridge event.
- You can subscribe to EventBridge events (such as this one) using AWS User Notifications.

```
{
    "version": "0",
    "id": "<id>", // alphanumeric string
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "<account ID>", // Management account ID.
    "time": "<date>", // Format: yyyy-MM-dd'T'hh:mm:ssZ
    "region": "<region>", // AWS Control Tower home region.
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "121212121212",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z", // Timestamp when call was made. Format:
 yyyy-MM-dd'T'hh:mm:ssZ.
        "eventSource": "controltower.amazonaws.com",
        "eventName": "<event name>", // one of the 9 event names in https://
docs.aws.amazon.com/controltower/latest/userguide/lifecycle-events.html
        "awsRegion": "<region>",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "<id>",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            // the contents of this object vary depending on the event subtype and
 event state
        }
    }
}
```

AWS Backup and AWS Control Tower

AWS Backup is a service that allows you to create plans for backing up your AWS resources automatically. To set up backups for your AWS Control Tower resources, you must follow four main steps:

1. Enable AWS Backup for your landing zone. You can do this on the **Landing zone settings** page in the AWS Control Tower console. When you turn on AWS Backup, resources are created in several accounts. For more information, see Resources created for AWS Backup.
2. Opt-in to backups for AWS Control Tower in the AWS Backup console. For more information, see Working with supported services in the *AWS Backup Developer Guide*.
3. Enable AWS Backup on the individual OUs you wish to include. You can do this task on the **OU details** page in the console, after you've enabled AWS Backup at the landing zone level. When you enable AWS Backup on an OU, the accounts in that OU receive local AWS Backup vaults.
4. Tag the selected resources to include in the backups. The tag denotes the frequency of backups for that resource. Your backup plan follows the schedule specified by the resource tags on each resource.

For more information, see *The AWS Backup Developer Guide*. No cost is incurred when you configure AWS Backup with AWS Control Tower. You will incur cost from AWS Backup. For information about pricing, see AWS Backup pricing.

For more details about the AWS Backup resources that AWS Control Tower creates in your AWS Control Tower landing zone, see Resources created for AWS Backup

Note

AWS Control Tower does not support setting up backup plans for AWS Control Tower resources through the AWS Backup service directly, without also enabling it in the AWS Control Tower service.

Prerequisites

Before you can set up AWS Backup for your AWS Control Tower resources, you must have an existing AWS Organizations organization. If you've already set up your AWS Control Tower landing zone, that serves as your existing organization.

You must allocate or create two other AWS accounts that are not enrolled in AWS Control Tower. These accounts become the *central backup account* and the *backup administrator account*. Name these accounts with those names.

Also, you must select or create a multi-Region AWS Key Management Service (KMS) key, specifically for AWS Backup.

Defining your prerequisites

- **The central backup account**—The central backup account stores your AWS Control Tower backup vault and your backups. This vault is created in all AWS Regions that AWS Control Tower governs, within this account. Cross-account copies are stored in this account, in case an account is compromised and requires data restoration.
- **The backup administrator account**—The backup administrator account is the delegated administrator account for the AWS Backup service in AWS Control Tower. It stores the Backup Audit Manager (BAM) report plans. This account aggregates all backup monitoring data, such as restore jobs and copy jobs. The data is stored in an Amazon S3 bucket. For more information, see Creating report plans using the AWS Backup console in the *AWS Backup Developer Guide*.
- **Policy requirement for the multi-Region AWS KMS key**

  Your AWS KMS key requires a key policy. Consider a key policy similar to this one, which restricts access to principals (users and roles) who have permissions associated with your organization's management account:

  JSON

```
{
    "Version": "2012-10-17",
    "Id": "KMS key policy",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the KMS key for organization",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey*",
                "kms:Encrypt",
                "kms:ReEncrypt*",
                "kms:GetKeyPolicy",
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "ORGANIZATION-ID"
                }
            }
        }
    ]
}
```

This example policy gives all accounts in your organization access to their encrypted backup data. Use AWS global condition keys and AWS KMS condition keys to refine the permissions, depending on which principals require access to your backups.

Note

Your multi-region AWS KMS key must be replicated for every AWS Region that you plan to govern with AWS Control Tower.

Enable backups

You can enable backups for resources in your accounts that are enrolled in AWS Control Tower, either during landing zone setup, or when you update your landing zone.

As Prerequisites, you must provide the following items

- An AWS account to serve as the AWS Backup Administrator account
- An AWS account to serve as the AWS Backup Central Backup account
- A multi-region AWS KMS key that you manage, for cross-account backups

How to enable backups

The enablement process has two main parts: *first*, enable backups for your landing zone; *then*, enable backups for each registered OU that requires backups.

First part: Set up backups for your landing zone

Console: You can set up backups for your landing zone in the AWS Control Tower console, on the **Landing zone settings** page. You'll see this option during the initial landing zone setup operation, and you can revisit it later with a landing zone update.

API: You can enable backups with the AWS Control Tower APIs, by calling the `UpdateLandingZone` API, if you already have an AWS Control Tower landing zone, or the `CreateLandingZone` API if you're setting up AWS Control Tower for the first time. (Hint: After that, call the `EnableBaseline` API to establish backups for each OU that you require.)

Outside the AWS Control Tower console

Part of enabling backups for your landing zone includes a step outside of the AWS Control Tower console. You must navigate to the AWS Backup console to review your resources.

To review your opted-in resource types or opt-in to additional resource types

1. Open the AWS Backup console at https://console.aws.amazon.com/backup.
2. In the navigation pane, choose **Settings**.
3. On the **Service opt-in** page, choose **Configure resources**.

4. Use the toggle switches to enable or disable the services that you want to include with AWS Backup. *Be sure that the resources that you want to back up are selected, such as RDS, EC2, DDB, and so forth, whether they are part of your AWS Control Tower environment or not.*

For more details, see Opt in to managing services with AWS Backup.

Considerations for new resource types

Before you rely on AWS Backup to manage data protection for any AWS service's resources, you must perform the previous procedure and opt in to AWS Backup for that service. Also, as the AWS Backup service adds support for additional services and their resource types in the future, you must repeat this procedure and opt in for each additional resource type with AWS Backup before you can back up that resource type in AWS Control Tower. Tagging an unsupported resource type may cause your backup to fail.

When you activate backups for your landing zone, AWS Control Tower establishes the two accounts you've provided as the **Central Backup** account and the **Backup Administrator** account, respectively. AWS Control Tower creates resources in these accounts and other accounts.

Important

To enable backups for the AWS Control Tower **Audit** and **Log Archive** accounts, you must set up backups for the **Security OU,** by calling the `EnableBaseline` API. We recommend that you do so.

The recommended bank of plans and retention is as follows:

- Hourly backups = 2-weeks retention in local vault, no copy in central backup vault
- Daily backups = 2-weeks retention in local vault, 1-month retention in central backup vault
- Weekly backups = 1-month retention in local vault, 3-months retention in central vault
- Monthly backups = 3-months retention in local vault, 3-months retention in central backup vault

For information about how to create your backup plans, see Creating report plans using the AWS Backup console.

Next part: Enable backups on OUs

After you enable AWS Backup in your landing zone settings, you must take the additional step to enable backup on the specific OUs that you want to back up. If you've enabled AWS Backup for your landing zone, you'll see a section on the **OU details** page in the console, which allows you to choose **Enable backup** for the OU. If backup is not enabled at the landing zone level, you will not see this section on the OU details page.

To enable the `BackupBaseline` on an OU, that OU must have the `AWSControlTowerBaseline` enabled already. Enrolled accounts in each OU have the `AWSControlTowerBaseline` enabled.

In your selected accounts and OUs, AWS Control Tower sets up additional resources

- **A local Backup vault**

  AWS Control Tower creates a local backup vault in your accounts, with four possible types of backup plans attached to the vault. Backup plans created through AWS Control Tower are tagged with a prefix.

  ```
  BackupPlanTags:
          aws-control-tower: 'managed-by-control-tower'
  ```

- **Four types of backup plans—hourly**, **daily**, **weekly**, **monthly**.

  Each plan is associated with a tag-based resource assignment. For example, any resource tagged with **aws-control-tower-backuphourly : true** is protected with an hourly backup plan.

- **A local backup role in your accounts**

  AWS Control Tower creates an IAM role, which is used for backups. The role requires four specific permissions.

  ```
  "backup:UpdateGlobalSettings","organizations:RegisterDelegatedAdministrator","organizations:E
  ```

  The role has a trust relationship with the service principal for AWS Backup The role is named `aws-controltower-backup-role`, and it has the following managed permissions attached to it:

  - [AWSBackupServiceRolePolicyForBackup]()
  - [AWSBackupServiceRolePolicyForRestores]()
  - [AWSBackupServiceRolePolicyForS3Backup]()

- [AWSBackupServiceRolePolicyForS3Restore](#)

Tag resources for backup

Part of the process of setting up backups in AWS Control Tower is to tag the resources that you wish to include in your backup plan. The tags specify the frequency of backups. These are the possible tags.

- `aws-control-tower-backuphourly : true`
- `aws-control-tower-backupdaily: true`
- `aws-control-tower-backupweekly: true`
- `aws-control-tower-backupmonthly: true`

Considerations

- When AWS Backup is active on an OU, you'll see a value of **Enabled** in the **Status** field on the **OU details** page in the AWS Control Tower console. Some other possible values of the **Status** field include **Not enabled**, **In progress**, and **Failed**. If you see a status of **Failed**, choose **Re-register OU** to reapply your AWS Backup configuration to the OU.
- If you have AWS Backup enabled on an OU, new accounts provisioned through Account Factory under that the OU will include AWS Backup.

Turn off backups

You can turn off backups for your resources in accounts that are enrolled in AWS Control Tower, either during landing zone setup, or when you update your landing zone.

Two main steps are required to turn off backups: first, turn off the AWS Backup baseline on each OU that has backups enabled, *then*, turn off backups for your landing zone.

First step: Turn off backups on OUs

If AWS Backup is enabled, you must disable the AWS Backup baseline from all OUs before you can turn off AWS Backup for your landing zone.

To disable the AWS Backup baseline on an OU, you can call the `DisableBaseline` API. The nested OUs inherit this status, so that the AWS Backup baseline baseline is disabled for them also.

Example command:

```
aws controltower disable-baseline --enabled-baseline-identifier Enabled-baseline-ARN
```

When you disable the the AWS Backup baseline, AWS Control Tower cleans up the following resources:

- All stacksets related to AWS Backup
- All controls related to AWS Backup

> **Note**
>
> The local vault is retained even though the stacksets are deleted, because the retention policy on the local vault is set to `Retain`. It preserves your data.

Next step: Turn off AWS Backup for your landing zone

After the prerequisite is met by turning off backups to your OUs, to turn off backups from the AWS Control Tower console, navigate to the **Landing zone settings** page. Choose **Disable backup**.

When you turn off AWS Backup, AWS Control Tower changes the following resources:

- Removes all stacksets related to AWS Backup
- Deactivates all controls related to AWS Backup in the Security OU
- De-registers the **Delegated admin** account for AWS Backup administration
- Removes AWS Control Tower governance (for CloudTrail, AWS Config, and so forth) from the AWS Backup Administrator and Central Backup accounts
- AWS Control Tower retains the AWS Backup vaults and Amazon S3 bucket resources containing your data

After you disable backups, no new backups are created, but existing backups are not removed.

Enable backup on moved accounts

If you move an account into an AWS Control Tower OU that has AWS Backup enabled, and the account is not enrolled in AWS Control Tower, your backup plan does not apply to the account automatically.

Console: To enable AWS Backup for an individual account from the AWS Control Tower console, you can choose **Update account** on the **Account details** page, or you can choose **Re-register OU** on the **OU detail**s page to update several accounts at the same time.

API: From the API, if you move an account into an OU that has the backup baseline enabled, you can call the `ResetEnabledBaseline` API on that OU, specifying the OU's `EnabledBaseline` resource as a target, to trigger backups on the account by inheritance from the OU.

Example command:

```
aws controltower reset-enabled-baseline --enabled-baseline-identifier
 arn:aws:controltower:REGION:NAMESPACE:enabledbaseline/XOSDORW8HDB5ZNWEE --region us-
east-1
```

Example response:

```
{
        "operationIdentifier": "0bbdb587-c849-4152-95c6-7afa7664ee71"
}
```

Backup drift in AWS Control Tower

Drift is not reported for AWS Backup configurations in AWS Control Tower. For more information about drift in AWS Control Tower, see Detect and resolve drift in AWS Control Tower.

If you delete or modify the AWS Backup plan, your plan can enter a state of drift. Here are some modifications to avoid.

- Do not move the **Backup Administrator** account from the Security OU.
- Do not move the **Central Backup** account from the Security OU.
- Do not remove the **Backup Administrator** account from the organization.
- Do not remove the **Central Backup** account from the organization.

- Do not detach, attach, or update the AWS Backup SCP that is applied to the Security OU.
- Do not detach, attach, or update the AWS Backup SCP that is applied to other OUs.
- Do not remove the **Backup Administrator** account's permission for AWS Backup.
- Do not update your cross-account backup settings to turn off cross-account backups. For more information about cross-account backups, see `UpdateGlobalSettings` in the *AWS Backup API Reference*.
- Do not delete your AWS KMS key.
- Do not modify your AWS KMS key policy after it is set.
- Do not disable the service's trusted access for AWS Backup.

Note

Drift is reported regarding the status of the SCP-based controls that protect AWS Backup resources in AWS Control Tower.

Resources created for AWS Backup

The tables on this page show resources that are created in AWS Control Tower accounts when you enable AWS Backup.

The following table shows the resources that AWS Control Tower creates in the AWS Control Tower **Central Backup** account when you enable AWS Backup for the landing zone organization.

Description	Resources for the Central Backup account
Which OU contains the account?	Security OU
What action created the resource?	Landing zone **Create** or **Update**
What resources are created?	Central Backup vault—`aws-controltower-central-backupvault-*`
What Regions are included?	All governed Regions
What controls are related to these resources?	`CT.BACKUP.PV.3`

The following table shows the resources that AWS Control Tower creates in the AWS Control Tower **Backup Administrator** account when you enable AWS Backup for the landing zone organization.

Description	Resources for the Backup Administrator account: This is the delegated administrator account for AWS Backup
Which OU contains the account?	Security OU
What action created the resource?	Landing zone **Create** or **Update**
What resources are created?	Backup Audit Manager (BAM) • `aws_controltower_copy_report` • `aws_controltower_backup_report` • `aws_controltower_restore_report` Amazon S3 bucket for storing BAM logs—`aws-controltower-backup-reports-` *`{accountId}`* `-*` Amazon S3 access logging bucket—`aws-controltower-backup-reports-log-` *`{accountId}`* `-*`
What Regions are included?	Home Region
What controls are related to these resources?	• `CT.BACKUP.PV.2` • `CT.S3.PV.1` • `CT.S3.PV.1`

The following table shows the resources that AWS Control Tower creates in the AWS Control Tower **Audit** account and in the AWS Control Tower **Log Archive** account when you enable AWS Backup for the Security OU.

Description	Resources for Audit and Log Archive accounts
Which OU contains the account?	Security OU
What action created the resource?	Enabling the `BackupBaseline`

What resources are created?	• Local Backup vault—`aws-controltower-local-backupvault-*`
	• Local Backup role—`aws-controltower-BackupRole`
	• Four local Backup plans (hourly, weekly, monthly, daily)
	  ◦ `aws-controltower-hourly-backup-plan`
	  ◦ `aws-controltower-daily-backup-plan`
	  ◦ `aws-controltower-weekly-backup-plan`
	  ◦ `aws-controltower-monthly-backup-plan`
	• An IAM role—`aws-controltower-BackupRole`

What Regions are included?	All governed Regions
What controls are related to these resources?	• `CT.BACKUP.PV.3` • `CT.IAM.PV.1` • `CT.BACKUP.PV.3` • `CT.BACKUP.PV.1`

> **Note**
>
> When you apply the `BackupBaseline` to the Security OU, all member accounts in that OU receive the AWS Backup resources, not just the **Audit** and **Log Archive** accounts.

The following table shows the resources that AWS Control Tower creates in the AWS Control Tower **OU member accounts** when you enable AWS Backup on a target OU.

Description	Resources for member accounts in other OUs
Which OU contains the account?	Any OU other than the Security OU
What action created the resource?	Enabling the `BackupBaseline`

What resources are created?	• Local Backup vault—`aws-controltower-local-backupvault-*`
	• Local Backup role—`aws-controltower-BackupRole`
	• Four local Backup plans (hourly, weekly, monthly, daily)
	  ◦ `aws-controltower-hourly-backup-plan`
	  ◦ `aws-controltower-daily-backup-plan`
	  ◦ `aws-controltower-weekly-backup-plan`
	  ◦ `aws-controltower-monthly-backup-plan`
	• An IAM role—`aws-controltower-BackupRole`

What Regions are included?	All governed Regions
What controls are related to these resources?	• `CT.BACKUP.PV.3` • `CT.IAM.PV.1`

Description	Resources for member accounts in other OUs
	• `CT.BACKUP.PV.3` • `CT.BACKUP.PV.1`

Controls for AWS Backup

When you enable AWS Backup in your AWS Control Tower landing zone, some preventive controls are activated in your environment. These controls protect the resources that AWS Backup needs to operate with AWS Control Tower. You cannot enable these controls if AWS Backup is not enabled for your landing zone.

For more information, see Controls for AWS Backup.

Decommission an AWS Control Tower landing zone

AWS Control Tower allows you to set up and govern secure multi-account AWS environments, known as landing zones. The process of cleaning up all of the resources allocated by AWS Control Tower is referred to as *decommissioning* a landing zone.

If you no longer want to use AWS Control Tower, the automated decommissioning tool cleans up the resources allocated by AWS Control Tower. To begin the automated decommissioning process, navigate to the **Landing Zone Settings** page, select the decommission tab, and choose **Decommission landing zone**.

For a list of actions performed during decommissioning, see Overview of the decommissioning process.

> **Warning**
>
> Manually deleting all of your AWS Control Tower resources is not the same as decommissioning. It will not allow you to set up a new landing zone.

Your data and your existing AWS Organizations are not changed by the decommissioning process, in the following ways.

- AWS Control Tower does not remove your data, it only removes parts of the landing zone that it created.
- After the decommissioning process is complete, a few resource artifacts remain, such as Amazon S3 buckets and Amazon CloudWatch Logs log groups. These resources must be deleted manually before you set up another landing zone, and to avoid possible costs associated with maintaining certain resources.
- You can't use automated decommissioning to remove a landing zone that's partially set up. If your landing zone setup process fails, you must resolve the failure state and set it up all the way to make automated decommissioning possible, or you must manually delete the resources individually.

Decommissioning a landing zone is a process with significant consequences, and it cannot be undone. The decommissioning actions taken by AWS Control Tower and the artifacts that remain after decommissioning are described in the following sections.

> **Important**
>
> We strongly recommend that you perform this decommissioning process only if you intend to stop using your landing zone. It is not possible to re-create your existing landing zone after you've decommissioned it.

Overview of the decommissioning process

When you request decommissioning of your landing zone, AWS Control Tower does the following actions.

- Disables each detective control enabled in the landing zone. AWS Control Tower deletes the AWS CloudFormation resources supporting the control.
- Disables each preventive control by removing service control policies (SCPs) from AWS Organizations. If a policy is empty (which it should be after removing all SCPs managed by AWS Control Tower), AWS Control Tower detaches and deletes the policy entirely.
- Deletes all blueprints deployed as AWS CloudFormation StackSets.
- Deletes all blueprints deployed as CloudFormation Stacks across all Regions.
- For each provisioned account, AWS Control Tower does the following actions during the decommissioning process.
  - Deletes records of each account factory account.
  - Revokes the AWS Control Tower permissions to the account by removing the IAM role that AWS Control Tower created (unless additional policies have been added to it) and recreates the standard `OrganizationsFullAccessRole` IAM role.
  - Removes records of the account from AWS Service Catalog.
  - Removes the account factory product and portfolio from AWS Service Catalog.
- Deletes the blueprints for the shared (Audit and Log Archive) accounts.
- Revokes the AWS Control Tower permissions from the shared accounts by removing the IAM role that AWS Control Tower created (unless additional policies have been added to it) and recreates the `OrganizationsFullAccessRole` IAM role.
- Deletes records related to the shared accounts.
- Deletes records related to customer-created OUs.
- Deletes internal records that identify the home Region.

Note

After decommissioning, you may wish to remove the Account Factory VPC blueprint (`BP_ACCOUNT_FACTORY_VPC`) to clean up the routes and NAT gateways, if your VPC was not empty.

How to decommission a landing zone

To decommission your AWS Control Tower landing zone from the console, follow the procedure given here.

Note

We recommend that you unmanage your enrolled accounts prior to decommissioning.

1. Navigate to the **Landing Zone Settings** page in the AWS Control Tower console.
2. Choose **Decommission your landing zone** within the **Decommission your landing zone** section.
3. A dialog appears, explaining the action you are about to perform, with a required confirmation process. To confirm your intent to decommission, you must select every box and type the confirmation as requested.

   Important

   The decommissioning process cannot be undone.
4. If you confirm your intent to decommission your landing zone, you are redirected to the AWS Control Tower home page while decommissioning is in progress. The process may require up to two hours.
5. When decommissioning has succeeded, you must delete remaining resources manually before setting up a new landing zone from the AWS Control Tower console. These remaining resources include some specific Amazon S3 buckets, organizations, and CloudWatch Logs log groups.

> **Note**
>
> *These actions may have significant consequences for your billing and compliance activities. For example, failure to delete these resources can result in unexpected charges.*

For more information about how to delete resources manually, see About removing AWS Control Tower resources.

6. If you intend to set up a new landing zone in a new AWS Region, follow this additional step. Enter the following command through the CLI:

```
aws organizations disable-aws-service-access --service-principal
 controltower.amazonaws.com
```

Decommission your landing zone with APIs

The process of cleaning up all of a landing zones resources is referred to as *decommissioning* a landing zone.

> **Important**
>
> We strongly recommend that you perform this decommissioning process only if you intend to stop using your landing zone. It is not possible to re-create your existing landing zone after you've decommissioned it.

For more details about decommissioning a landing zone, including important information about how AWS Control Tower handles your data and existing AWS Organizations, review Decommission an AWS Control Tower landing zone.

To decommission a landing zone, call `DeleteLandingZone` API. This API returns an `OperationIdentifier`, which you can then use when calling the `GetLandingZoneOperation` API to check the delete operation's status.

```
aws controltower delete-landing-zone --landing-zone-identifier
"arn:aws:controltower:us-west-2:123456789012:landingzone/1A2B3C4D5E6F7G8H"
```

Output:

```
{
   "operationIdentifier": "55XXXXXX-e2XX-41XX-a7XX-446XXXXXXXXX"
}
```

Manual cleanup tasks required after decommissioning

This section lists manual cleanup tasks you must perform after the initial decommissioning step.

- You must specify different email addresses for the Log archive and Audit accounts if you create a new landing zone after decommissioning one, or follow the procedure for bringing your own existing Log archive or Audit accounts.
- The CloudWatch Logs log group, `aws-controltower/CloudTrailLogs`, must be deleted manually before you set up another landing zone.
- The two Amazon S3 buckets with reserved names for logs must be removed, or renamed, manually.
- You must delete, or rename, the existing **Security** and **Sandbox** organizational units manually.

> **Note**
>
> Before you can delete the AWS Control Tower **Security OU** organization, you must first delete the logging and audit accounts, but not the management account. To delete these accounts, you must When to sign in as a root user to the audit account and to the logging account and delete them individually.

- You may wish to delete the AWS IAM Identity Center (IAM Identity Center) configuration for AWS Control Tower manually, but you can proceed with the existing IAM Identity Center configuration.
- You may wish to remove the VPC created by AWS Control Tower, and remove the associated AWS CloudFormation stack set.
- Before you can set up a new landing zone in a new AWS Region, you must follow these additional steps.
  - Enter the following command through the CLI:

```
aws organizations disable-aws-service-access --service-principal
 controltower.amazonaws.com
```

- Delete the remaining managed rule, called `AWSControlTowerManagedRule`, from the shared and member accounts for all governed Regions. `AWSControlTowerManagedRule` is an Amazon EventBridge rule.

Resources not removed during decommissioning

Decommissioning a landing zone does not fully reverse the AWS Control Tower setup process. Certain resources remain, which may be removed manually.

AWS Organizations

For customers without existing AWS Organizations organizations, AWS Control Tower sets up an organization with two organizational units (OUs), named **Security** and **Sandbox**. When you decommission your landing zone, the hierarchy of the organization is preserved, as follows:

- Organizational Units (OUs) you created from the AWS Control Tower console are not removed.
- The Security and Sandbox OUs are not removed.
- The organization is not deleted from AWS Organizations.
- No accounts in AWS Organizations (shared, provisioned, or management) are moved or removed.

AWS IAM Identity Center (SSO)

For customers without an existing IAM Identity Center directory, AWS Control Tower sets up IAM Identity Center and configures an initial directory. When you decommission your landing zone, AWS Control Tower makes no changes to IAM Identity Center. If needed, you can delete the IAM Identity Center information stored in your management account manually. In particular, these areas are unchanged by decommissioning:

- Users created with Account Factory are not removed.
- Groups created by AWS Control Tower setup are not removed.
- Permission sets created by AWS Control Tower are not removed.
- Associations between AWS accounts and IAM Identity Center permission sets are not removed.

- IAM Identity Center directories are not changed.
- These IAM Identity Center policies for AWS Control Tower are not removed:
  - `AWSControlTowerAdminPolicy`
  - `AWSControlTowerCloudTrailRolePolicy`
  - `AWSControlTowerStackSetRolePolicy`

Roles

During setup, AWS Control Tower creates certain roles for you if you use the console, or it asks you to create these roles if you set up your landing zone through the APIs. When you decommission your landing zone, the following roles are not removed:

- `AWSControlTowerAdmin`
- `AWSControlTowerCloudTrailRole`
- `AWSControlTowerStackSetRole`
- `AWSControlTowerConfigAggregatorRoleForOrganizations`

Amazon S3 Buckets

During setup, AWS Control Tower creates buckets in the logging account for logging and for logging access. When you decommission your landing zone, the following resources are not removed:

- Logging and logging access S3 buckets in the logging account are not removed.
- Contents of the logging and logging access buckets are not removed.

Shared Accounts

Two shared accounts (Audit and Log Archive) are created in the Security OU during AWS Control Tower setup. When you decommission your landing zone:

- Shared accounts that were created during AWS Control Tower setup are not closed.
- The `OrganizationAccountAccessRole` IAM role is recreated to align with standard AWS Organizations configuration.
- The `AWSControlTowerExecution` role is removed.

Provisioned Accounts

AWS Control Tower customers can use account factory to create new AWS accounts. When you decommission your landing zone:

- Provisioned accounts you created with Account Factory are not closed.
- Provisioned products in AWS Service Catalog are not removed. If you clean those up by terminating them, their accounts are moved into the **Root OU**.
- The VPC that AWS Control Tower created is not removed, and the associated AWS CloudFormation stack set (`BP_ACCOUNT_FACTORY_VPC`) is not removed.
- The `OrganizationAccountAccessRole` IAM role is recreated to align with standard AWS Organizations configuration.
- The `AWSControlTowerExecution` role is removed.

CloudWatch Logs Log Group

A CloudWatch Logs log group, `aws-controltower/CloudTrailLogs`, is created as part of the blueprint named `AWSControlTowerBP-BASELINE-CLOUDTRAIL-MANAGEMENT`. This log group is not removed. Instead, the blueprint is deleted and the resources are retained.

- This log group must be deleted manually before you set up another landing zone.

> **Note**
>
> Customers on landing zone 3.0 and later do not need to delete their individual enrolled account's CloudTrail logs and CloudTrail logs roles, because these are created in the management account only, for the organization-level trail.
> Beginning with landing zone version 3.2, AWS Control Tower creates an Amazon EventBridge rule, called `AWSControlTowerManagedRule`. This rule is created in each member account, for all governed Regions. The rule is not deleted automatically during decommissioning, so you must delete it manually from the shared and member accounts for all governed Regions before you can set up a landing zone in a new Region.

Procedures for how to delete AWS Control Tower resources are given in [Remove AWS Control Tower resources](#).

Remove AWS Control Tower resources

This document provides instructions for how to remove AWS Control Tower resources individually, as part of regular maintenance and administrative tasks. The procedures given in this chapter are intended only for removing individual resources, or a few resources, when needed. It not the same as decommissioning your landing zone.

Two types of tasks may require you to remove resources:

- To delete resources as you manage your landing zone in ordinary situations.
- To clean up resources that remain after automated decommissioning.

> **Warning**
>
> Manually removing resources will not allow you to set up a new landing zone. It is not the same as decommissioning. If you intend to decommission your AWS Control Tower landing zone, follow the instructions on Decommission an AWS Control Tower landing zone before you take any actions described in this chapter. The instructions in this chapter can help you clean up resources that remain after automated decommissioning is complete. Even if you delete all of your landing zone resources manually, it is not the same as decommissioning the landing zone, and you may incur unexpected charges.

If you need to remove an account from AWS Control Tower, see the following sections to close an account:

- Unmanage an account
- Close an account created in Account Factory

Do I need decommissioning instead of deleting?

If you no longer intend to use AWS Control Tower for your enterprise, or if you require a major redeployment of your organizational resources, you may want to decommission the resources created when you initially set up your landing zone.

- After the decommissioning process is complete, a few resource artifacts remain, such as Amazon S3 buckets and Amazon CloudWatch Logs log groups.

- You must clean up the remaining resources in your accounts manually before you set up another landing zone, and to avoid the possibility of unexpected charges. For more information, see Resources not removed during decommissioning.

> **Warning**
>
> We strongly recommend that you perform a decommissioning process *only if* you intend to stop using your landing zone. This process cannot be undone.

About removing AWS Control Tower resources

The individual procedures in this chapter guide you through manual methods of removing AWS Control Tower resources. These procedures can be followed when you need to delete a specific resource from your landing zone.

Before performing these procedures, unless it's otherwise indicated, you must be signed in to the AWS Management Console in the home Region for your landing zone, and you must be signed in as an IAM user or user in IAM Identity Center with administrative permissions for the management account that contains your landing zone.

> **Warning**
>
> These are destructive actions that can introduce governance drift into your AWS Control Tower setup. They cannot be undone.

Topics

- Delete SCPs
- Delete StackSets and Stacks
- Delete Amazon S3 Buckets in the Log Archive Account
- Remove an Account Factory Portfolio and Product
- Remove AWS Control Tower Roles and Policies
- AWS Control Tower resource help

Delete SCPs

AWS Control Tower uses service control policies (SCPs) for its controls. This procedure walks through how to delete the SCPs specifically related to AWS Control Tower.

To delete AWS Organizations SCPs

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
2. Open the **Policies** tab, and find the Service Control Policies (SCPs) that have the prefix **aws-guardrails-** and do the following for each SCP:

   a. Detach the SCP from the associated OU.

   b. Delete the SCP.

Delete StackSets and Stacks

AWS Control Tower uses StackSets and stacks to deploy AWS Config Rules related to controls in your landing zone. The following procedures walk through how to delete these specific resources.

To delete AWS CloudFormation StackSets

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. From the left navigation menu, choose **StackSets**.
3. For each StackSet with the prefix **AWSControlTower**, do the following. If you have many accounts in a StackSet, this can take some time.

   a. Choose the specific StackSet from the table in the dashboard. This opens the properties page for that StackSet.

   b. At the bottom of the page, in the **Stacks** table, make a record of the AWS account IDs for all the accounts in the table. Copy the list of all accounts.

   c. From **Actions**, choose **Delete stacks from StackSet**.

   d. On **Set deployment options**, from **Deployment locations**, choose **Deploy stacks in accounts**.

   e. In the text field, enter the AWS account IDs you made a record of in step 3.b, separated by commas. For example: *123456789012*, *098765431098*, and so on.

   f. From **Specify regions**, choose **Add all**, leave the rest of the parameters on the page set to their defaults, and choose **Next**.

g. On the **Review** page, review your choices, and then choose **Delete stacks**.

h. On the **StackSet properties** page, you can begin this procedure again for your other StackSets.

4. The process is complete when the records in the **Stacks** table of the different **StackSets properties** pages are empty.
5. When the records in the **Stacks** table are empty, choose **Delete StackSet**.

To delete AWS CloudFormation stacks

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. From the **Stacks** dashboard, search for all of the stacks with the prefix **AWSControlTower**.
3. For each stack in the table, do the following:

   a. Choose the check box next to the name of the stack.

   b. From the **Actions** menu, choose **Delete Stack**.

   c. In the dialog box that opens, review the information to make sure it's accurate, and choose **Yes, Delete**.

Delete Amazon S3 Buckets in the Log Archive Account

The following procedures guide you through how to sign in to the log archive account as an IAM Identity Center user in the **AWSControlTowerExecution** group and then delete the Amazon S3 buckets in your log archive account.

To sign in to your log archive account with the right permissions

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
2. From the **Accounts** tab, find the **Log archive** account.
3. From the right pane that opens, make a record of the log archive account number.
4. From the navigation bar, choose your account name to open your account menu.
5. Choose **Switch Role**.
6. On the page that opens, provide the account number for the log archive account in **Account**.
7. For **Role**, enter **AWSControlTowerExecution**.
8. The **Display Name** populates with text.

9. Choose your favorite **Color**.
10. Choose **Switch Role**.

To delete Amazon S3 buckets

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Search for bucket names that contain **aws-controltower**.
3. For each bucket in the table, do the following:
   a. Choose the check box for the bucket in the table.
   b. Choose **Delete**.
   c. In the dialog box that opens, review the information to make sure it's accurate, enter the name of the bucket to confirm, and then choose **Confirm**.

Remove an Account Factory Portfolio and Product

The following procedure guides you through how to sign in as an IAM Identity Center user in the **AWSServiceCatalogAdmins** group and then clean up your Account Factory portfolio and products.

To sign in to your management account with the right permissions

1. Go to your user portal URL at *`directory-id`*.awsapps.com/start
2. From **AWS Account**, find the **Management** account.
3. From **AWSServiceCatalogAdminFullAccess**, choose **Management console** to sign in to the AWS Management Console as this role.

To clean up Account Factory

1. Open the Service Catalog console at https://console.aws.amazon.com/servicecatalog/.
2. From the left navigation menu, choose **Portfolios list**.
3. In the **Local Portfolios** table, search for a portfolio named **AWS Control Tower Account Factory Portfolio**.
4. Choose the name of that portfolio to go to its details page.
5. Expand the **Constraints** section of the page, and choose the radio button for the constraint with the product name **AWS Control Tower Account Factory**.

6. Choose **REMOVE CONSTRAINTS**.
7. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
8. From the **Products** section of the page, choose the radio button for the product named **AWS Control Tower Account Factory**.
9. Choose **REMOVE PRODUCT**.
10. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
11. Expand the **Users, Groups, and Roles** section of the page, and choose the check boxes for all the records in this table.
12. Choose **REMOVE USERS, GROUP OR ROLE**.
13. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
14. From the left navigation menu, choose **Portfolios list**.
15. In the **Local Portfolios** table, search for a portfolio named **AWS Control Tower Account Factory Portfolio**.
16. Choose the radio button for that portfolio, and then choose **DELETE PORTFOLIO**.
17. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
18. From the left navigation menu, choose **Product list**.
19. On the **Admin products** page, search for the product named **AWS Control Tower Account Factory**.
20. Choose the product to open the **Admin product details** page.
21. From **Actions**, choose **Delete product**.
22. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.

Remove AWS Control Tower Roles and Policies

These procedures walk you through how to clean up the roles and policies that AWS Control Tower created when your landing zone was set up, or later.

To delete the IAM Identity Center AWSServiceCatalogEndUserAccess role

1. Open the AWS IAM Identity Center console at https://console.aws.amazon.com/singlesignon/.
2. Change your AWS Region to your home Region, which is the Region where you initially set up AWS Control Tower.
3. From the left navigation menu, choose **AWS accounts**.
4. Choose your management account link.
5. Choose the dropdown for **Permission sets**, select **AWSServiceCatalogEndUserAccess**, and then choose **Remove**.
6. Choose **AWS accounts** from the left panel.
7. Open the **Permission sets** tab.
8. Select **AWSServiceCatalogEndUserAccess** and delete it.

To delete IAM roles

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. From the left navigation menu, choose **Roles**.
3. From the table, search for roles with the name **AWSControlTower**.
4. For each role in the table, do the following:

    a. Choose the check box for the role.

    b. Choose **Delete role**.

    c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Yes, delete**.

To delete IAM policies

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. From the left navigation menu, choose **Policies**.
3. From the table, search for policies with the name **AWSControlTower**.
4. For each policy in the table, do the following:

    a. Choose the check box for the policy.

    b. Choose **Policy actions**, and **Delete** from the dropdown menu.

c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Delete**.

AWS Control Tower resource help

If you encounter any issues that you can't resolve when you remove AWS Control Tower resources, contact AWS Support.

Setup after decommissioning a landing zone

After you decommission your landing zone, you cannot successfully execute setup again until manual cleanup is complete. Also, without manual cleanup of these remaining resources, you may incur unexpected billing charges. You must attend to these issues:

- The AWS Control Tower management account is part of the AWS Control Tower **Root OU**. Be sure that these IAM roles and IAM policies are removed from the management account:
  - Roles:

    ```
    - AWSControlTowerAdmin

    - AWSControlTowerCloudTrailRole

    - AWSControlTowerStackSetRole
    ```
  - Policies:

    ```
    - AWSControlTowerAdminPolicy

    - AWSControlTowerCloudTrailRolePolicy

    - AWSControlTowerStackSetRolePolicy
    ```
- You may wish to delete or update the existing IAM Identity Center configuration for AWS Control Tower before you set up a landing zone again, but it is not required that you delete it.
- You may wish to remove the VPC created by AWS Control Tower.
- Setup fails if the email addresses specified for the logging or audit accounts are associated with an existing AWS account. You may close the AWS accounts, or use different email addresses to set up a landing zone again. Alternatively, you may re-use these existing shared accounts, with

the feature that allows you to bring your own logging and audit accounts. For more information, see Considerations for bringing existing security or logging accounts.

- Setup fails if Amazon S3 buckets with the following reserved names already exist in the logging account:
  - `aws-controltower-logs-{accountId}-{region}` (used for the logging bucket).
  - `aws-controltower-s3-access-logs-{accountId}-{region}` (used for the logging access bucket).

  You must either rename or remove these buckets, or use a different account for the logging account.
- Setup fails if the management account has the existing log group, `aws-controltower/CloudTrailLogs`, in CloudWatch Logs. You must either rename or remove the log group.

Before you set up in a new AWS Region

If you intend to set up a new landing zone in a new AWS Region, follow these additional steps.

- Enter the following command through the CLI:

```
aws organizations disable-aws-service-access --service-principal
 controltower.amazonaws.com
```

- Delete the remaining managed rule, called `AWSControlTowerManagedRule`, from shared and member accounts for all governed Regions.

> **Note**
>
> You cannot set up a new landing zone in an organization with top-level OUs named either **Security** or **Sandbox**. You must rename or remove these OUs to set up a landing zone again.

Walkthroughs

This chapter contains walkthrough procedures that can help you in your use of AWS Control Tower.

Topics

Walkthrough: Move from ALZ to AWS Control Tower

Many AWS customers have adopted the AWS Landing Zone solution (ALZ) to set up a secure, compliant, multi-account AWS environment. To reduce the burden of managing a landing zone, AWS created the managed service called AWS Control Tower.

No additional features are scheduled for ALZ; it is in long-term support only. Therefore, we recommend that you move to the AWS Control Tower service from ALZ. The blog that is linked in this chapter walks you through different considerations for that move, and it explains how you can plan a successful migration from ALZ to AWS Control Tower.

Blog: Migrate AWS Landing Zone solution to AWS Control Tower

AWS Prescriptive Guidance offers more extensive documentation, including steps for transitioning from ALZ to AWS Control Tower. Essentially, you will enable AWS Control Tower governance in your existing organization that is running ALZ, based upon a number of prerequisites. For information, see Transitioning from AWS Landing Zone to AWS Control Tower.

Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs

AWS Control Tower is integrated with several other AWS services, such as AWS Service Catalog. You can use the APIs to create and provision your member accounts in AWS Control Tower.

The video shows you how to provision accounts in an automated, batch fashion, by calling the AWS Service Catalog APIs. For provisioning, you'll call the `ProvisionProduct` API from the AWS command line interface (CLI), and you'll specify a JSON file that contains the parameters for each account you'd like to set up. The video illustrates installing and using the AWS Cloud9 development environment to perform this work. The CLI commands would be the same if you use AWS Cloudshell instead of AWS Cloud9.

> **Note**
>
> You also can adapt this approach for automating account updates, by calling the `UpdateProvisionedProduct` API of AWS Service Catalog for each account. You can write a script to update the accounts, one by one.

As a completely different automation method, if you are familiar with Terraform, you can provision accounts with AWS Control Tower Account Factory for Terraform (AFT).

Sample automation administration role

Here is a sample template you can use to help configure your automation administration role in the management account. You would configure this role in your management account so it can perform the automation with Administrator access in the target accounts.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure the SampleAutoAdminRole

Resources:
  AdministrationRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: SampleAutoAdminRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: cloudformation.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      Policies:
```

```
        - PolicyName: AssumeSampleAutoAdminRole
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - sts:AssumeRole
                Resource:
                  - "arn:aws:iam::*:role/SampleAutomationExecutionRole"
```

Sample automation execution role

Here is a sample template you can use to help you set up your automation execution role. You would configure this role in the target accounts.

```
AWSTemplateFormatVersion: "2010-09-09"
Description: "Create automation execution role for creating Sample Additional Role."

Parameters:
  AdminAccountId:
    Type: "String"
    Description: "Account ID for the administrator account (typically management,
 security or shared services)."
  AdminRoleName:
    Type: "String"
    Description: "Role name for automation administrator access."
    Default: "SampleAutomationAdministrationRole"
  ExecutionRoleName:
    Type: "String"
    Description: "Role name for automation execution."
    Default: "SampleAutomationExecutionRole"
  SessionDurationInSecs:
    Type: "Number"
    Description: "Maximum session duration in seconds."
    Default: 14400

Resources:
  # This needs to run after AdminRoleName exists.
  ExecutionRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: !Ref ExecutionRoleName
      MaxSessionDuration: !Ref SessionDurationInSecs
```

```
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              AWS:
                - !Sub "arn:aws:iam::${AdminAccountId}:role/${AdminRoleName}"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      ManagedPolicyArns:
        - "arn:aws:iam::aws:policy/AdministratorAccess"
```

After configuring these roles, you call the AWS Service Catalog APIs to perform the automated tasks. The CLI commands are given in the video.

Sample provisioning input for Service Catalog API

Here is a sample of the input you can give to the Service Catalog `ProvisionProduct` API if you're using the API to provision AWS Control Tower accounts:

```
{
  pathId: "lpv2-7n2o3nudljh4e",
  productId: "prod-y422ydgjge2rs",
  provisionedProductName: "Example product 1",
  provisioningArtifactId: "pa-2mmz36cfpj2p4",
  provisioningParameters: [
    {
      key: "AccountEmail",
      value: "abc@amazon.com"
    },
    {
      key: "AccountName",
      value: "ABC"
    },
    {
      key: "ManagedOrganizationalUnit",
      value: "Custom (ou-xfe5-a8hb8ml8)"
    },
    {
      key: "SSOUserEmail",
      value: "abc@amazon.com"
    },
```

```
    {
      key: "SSOUserFirstName",
      value: "John"
    },
    {
      key: "SSOUserLastName",
      value: "Smith"
    }
  ],
  provisionToken: "c3c795a1-9824-4fb2-a4c2-4b1841be4068"
}
```

For more information, see the [API reference for Service Catalog](https://docs.aws.amazon.com/servicecatalog/).

> **Note**
>
> Notice that the format of the input string for the value of `ManagedOrganizationalUnit` has changed from `OU_NAME` to `OU_NAME (OU_ID)`. The video that follows does not mention this change.

Video Walkthrough

This video (6:58) describes how to automate account deployments in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Automated Account Provisioning in AWS Control Tower.

Walkthrough: Configure AWS Control Tower Without a VPC

This topic walks through how to configure your AWS Control Tower accounts without a VPC.

If your workload does not require a VPC, you can do the following:

- You can delete the AWS Control Tower virtual private cloud (VPC). This VPC was created when you set up your landing zone.
- You can change your Account Factory settings so that new AWS Control Tower accounts are created without an associated VPC.

> **Important**
>
> If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides the control Disallow internet access for an Amazon VPC instance managed by a customer. To avoid enabling internet access for newly provisioned accounts, you must change the setting in Account Factory.

Delete the AWS Control Tower VPC

Outside of AWS Control Tower, every AWS customer has a default VPC, which you can view on the Amazon Virtual Private Cloud (Amazon VPC) console at https://console.aws.amazon.com/vpc/. You'll recognize the default VPC, because its name always includes the word *(default)* at the end of the name.

When you set up a AWS Control Tower landing zone, AWS Control Tower deletes your AWS default VPC and creates a new AWS Control Tower default VPC. The new VPC is associated with your AWS Control Tower management account. This topic refers to that new VPC as the *Control Tower VPC*.

When you view your AWS Control Tower VPC in the Amazon VPC console, you will *not* see the word *(default)* at the end of the name. If you have more than one VPC, you must use the assigned CIDR range to identify the correct AWS Control Tower VPC.

You can delete the AWS Control Tower VPC, but if you later need a VPC in AWS Control Tower, you must create it yourself.

To delete the AWS Control Tower VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Search for **VPC** or select **VPC** from the Service Catalog options. You then see the **VPC Dashboard**.
3. From the menu on the left, choose **Your VPCs**. You then see a list of all your VPCs.
4. Identify the AWS Control Tower VPC by its CIDR range.
5. To delete the VPC, choose **Actions** and then choose **Delete VPC**.

An AWS *(default)* VPC already exists in every Region for the AWS Control Tower management account. To follow security best practices, if you choose to delete the AWS Control Tower VPC, it's best also to delete the AWS default VPC associated with the management account from all AWS Regions. Therefore, to secure the management account, remove the default VPC from each Region, as well as removing the VPC created by Control Tower in your AWS Control Tower home region.

Optionally clean up the VPC resource in the account

Optionally, to clean up the AWS Control Tower VPC resource, `aws-controltower-VPC`, from an existing account, you can remove the stack instance from the AWS CloudFormation StackSet `AWSControlTowerBP-VPC-ACCOUNT-FACTORY-V1`, after you make sure that there are no existing resources or resource dependencies in the VPC.

Create an Account in AWS Control Tower Without a VPC

If your end-user workloads do not require VPCs, you can use this method to set up end-user accounts that don't have VPCs created for them automatically.

From the AWS Control Tower dashboard, you can view and edit your network configurations settings. After you change the settings so that AWS Control Tower accounts are created without an associated VPC, all new accounts are created without a VPC until you change the settings again.

To configure Account Factory for creating accounts without VPCs

1. Open a web browser, and navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. Choose **Account Factory** from the menu on the left.
3. You then see the Account Factory page with the **Network Configuration** section.
4. Note the current settings if you intend to restore them later.
5. Choose the **Edit** button in the **Network Configuration** section.
6. In the **Edit account factory network configuration** page, go to the **VPC Configuration options for new accounts** section.

   You can follow **Option 1** or **Option 2**, or both, to ensure that AWS Control Tower does not create a VPC when provisioning an account.

   a. **Option 1 – Removing subnets**

      - Turn off the **Internet-accessible subnet** toggle switch.
      - Set the **Maximum number of private subnets** value to 0.

   b. **Option 2 – Removing AWS Regions**

      - Clear every checkbox in the **Regions for VPC creation** column.

7. Choose **Save**.

Possible Errors

Be aware of these possible errors that could occur when you delete your AWS Control Tower VPC or reconfigure Account Factory to create accounts without VPCs.

- Your existing management account may have dependencies or resources in the AWS Control Tower VPC, which can cause a *deletion failure* error.
- If you leave the default CIDR in place when setting up to launch new accounts without a VPC, your request fails with an error that *the CIDR is not valid*.

Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager

The video shows you how to use the AWS Firewall Manager service to provide improvements to your network security for AWS Control Tower. You can designate a security administrator account that's enabled to set up security groups. You will see how you can configure security policies and enforce security rules for your AWS Control Tower organizations, and how you can remediate non-compliant resources by applying policies automatically. You can view the security groups that are in effect for each account and resource (such as an Amazon EC2 instance) in your organization.

You can create your own firewall policies, or you can subscribe to rules from trusted vendors.

Set Up Security Groups With AWS Firewall Manager

This video (8:02) describes how to set up better network infrastructure security for your resources and workloads in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Firewall Setup in AWS Control Tower.

For more information, see the documentation on how to set up AWS WAF.

Troubleshooting

If you encounter issues while using AWS Control Tower, you can use the following information to resolve them according to our best practices. If the issues you encounter are outside the scope of the following information, or if they persist after you've tried to resolve them, contact AWS Support.

Topics

Landing Zone Launch Failed

Common causes of landing zone launch failure:

- Lack of response to a confirmation email message.
- AWS CloudFormation StackSet failure.

Confirmation email messages: If your management account is less than an hour old, you may encounter issues when the additional accounts are created.

Action to take

If you encounter this issue, check your email. You might have been sent confirmation email that is awaiting response. Alternatively, we recommend that you wait an hour, and then try again. If the issue persists, contact AWS Support.

Failed StackSets: Another possible cause of landing zone launch failure is AWS CloudFormation StackSet failure. AWS Security Token Service (STS) regions must be enabled in the management account for all AWS Regions that AWS Control Tower is governing, so that the provisioning can be successful; otherwise, stack sets will fail to launch.

Action to take

Be sure to enable all of your required AWS Security Token Service (STS) endpoint regions before you launch AWS Control Tower.

To remediate failed stack sets and retry the setup

1. Navigate to the AWS CloudFormation console in the appropriate AWS Region.
2. Clean up the failed stack with the name **AWSControlTowerBP-BASELINE-CLOUDTRAIL-MASTER**.
3. Wait for the stack deletion to complete.
4. Return to the AWS Control Tower page.
5. Select **Set up landing zone** again.

To view a list of AWS Regions that AWS Control Tower supports, see How AWS Regions Work With AWS Control Tower.

Landing Zone Update Failed with KMS Error

The most common cause of landing zone update failure is a AWS KMS key policy that is not valid.

Common reasons for a non-valid KMS key policy:

- Typo in the policy.
- Forgot to add the required policy statement.
- Incorrect AWS partition.

- Incorrect account number in the policy.
- Forgot to remove the placeholders in the example policy.

Action to take

Review your policy to check for these errors.

For more information about AWS KMS key policies, see Configure KMS keys and Backup prerequisites.

Unable to Update Landing Zone

AWS Control Tower does not roll back to a previous landing zone version if an update fails. You may find your landing zone in an indeterminate state. If so, contact AWS support.

Landing zone updates may fail for several reasons.

- Prerequisites not met
- AWS Config resources exist in certain accounts
- Closed accounts exist

Prerequisites not met

A landing zone update must meet the same prerequisites as a landing zone setup. Before you update, review the pre-launch checks.

AWS Config resources exist in Security OU accounts

Do not add AWS Config resources in your **Audit** and **Log archive** accounts. The landing zone update process cannot complete with these resources present. These restrictions are similar to those for enrolling an account or setting up a landing zone for the first time. For more information, see Enroll accounts that have existing AWS Config resources.

Closed accounts exist

When an account is in a **Closed** or **Suspended** state, you may encounter an issue when you try to update your landing zone. You must delete the provisioned product on every closed account before you perform an update to the landing zone.

On the AWS Service Catalog provisioned product page, you may see an error message similar to this one:

```
AWSControlTowerExecution role can't be assumed on the account.
```

Common cause: You have suspended an account without deleting the provisioned product.

Action to take: If you see this error, you have two options:

1. Contact AWS Support and reopen the account, delete the provisioned product, then close the account again.
2. Remove the resources from the StackSets that have been orphaned because of the account closure. (This option is available only if the StackSets have instances in **Current** state that you are not removing.)

To remove the resources from the StackSets, do this for each closed account:

- Go into each of the AWS Control Tower StackSets and remove the StackInstances from every region, for the account that has been closed.
- **IMPORTANT:** Choose the **Retain Stack** option so the StackSet removes only the stack instances. StackSet can't assume a role from the closed account, so it will fail if it tries to assume the `AWSControlTowerExecution` role, which leads to the error message you received.

Landing zone not up to date error

If you have not updated your landing zone recently, you may receive an error when you try to regain access to AWS Control Tower. You may see an error message similar to this one:

```
Unable to access Control Tower
```

Your account has been inactive for too long. Due to inactivity, you must update your landing zone for access to AWS Control Tower.

However, your landing zone update may fail.

Steps to take

Sign in to the management account of your organization, and sign in as root user. Your IAM user or user in IAM Identity Center must have AWS Control Tower administrator permissions and be part of the **AWSControlTowerAdmins** group. Then try the update again.

New Account Provisioning Failed

If you encounter this issue, check for these common causes.

When you filled out the account provisioning form, you may have:

- specified **tagOptions**,
- enabled SNS notifications,
- enabled provisioned product notifications.

Try again to provision your account, without specifying any of those options. For more information, see Provision accounts with AWS Service Catalog Account Factory .

Other common causes for failure:

- If you created a provisioned product plan (to view resource changes), your account provisioning may remain in an **In progress** state indefinitely.
- Creation of a new account in Account Factory will fail while other AWS Control Tower configuration changes are in progress. For example, while a process is running to add a control to an OU, Account Factory will display an error message if you try to provision an account.

To check the status of a previous action in AWS Control Tower

- Navigate to **AWS CloudFormation > StackSets**
- Check each stack set related to AWS Control Tower (prefix: "AWSControlTower")
- Look for AWS CloudFormation StackSets operations that are still running.

If your account provisioning takes longer than one hour, it's best to terminate the provisioning process and try again.

Failed to Enroll an Existing Account

If you try once to enroll an existing AWS account and that enrollment fails, when you try a second time, the error message may tell you that the stack set exists. To continue, you must remove the provisioned product in Account Factory.

If the reason for the first enrollment failure was that you forgot to create the `AWSControlTowerExecution` role in the account in advance, the error message you'll receive correctly tells you to create the role. However, when you try to create the role, you are likely to receive another error message stating that AWS Control Tower could not create the role. This error occurs because the process has been partially completed.

In this case, you must take two recovery steps before you can proceed with enrolling your existing account. First, you must terminate the Account Factory provisioned product through the AWS Service Catalog console. Next, you must use the AWS Organizations console to manually move the account out of the OU and back to the root. After that is done, create the `AWSControlTowerExecution` role in the account, and then fill in the **Enroll account** form again.

Another possible cause of enrollment failure is that the account has existing AWS Config resources. In that case, see Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.

Unable to Update an Account Factory Account

When an account is in an inconsistent state, it cannot be updated successfully from Account Factory or AWS Service Catalog.

Case 1: You may encounter an error message similar to this one:

```
AWS Control Tower could not baseline VPC in the managed account because of
existing resource dependencies.
```

Common cause: AWS Control Tower always removes the AWS default VPC during initial provisioning. To have an AWS default VPC in an account, you must add it after account creation. AWS Control Tower has its own default VPC that replaces the AWS default VPC, unless you set up Account Factory the way the walkthrough shows you—-so that AWS Control Tower doesn’t provision a VPC at all. Then the account has no VPC. You’d have to re-add the AWS default VPC if you want to use that one.

However, AWS Control Tower doesn't support the AWS default VPC. Deploying one causes the account to enter a `Tainted` state. When it is in that state, you cannot update the account through AWS Service Catalog.

Action to take: You must delete the default VPC that you added, and then you will be able to update the account.

> **Note**
>
> The `Tainted` state causes a follow-on issue: An account that is not updated may prevent enabling controls on the OU of which it is a part.

Case 2: You may see an error message similar to this one:

```
AWS Control Tower detects that your enrolled account has been moved to a
new organizational unit.
```

Common cause: You attempted to move an account from one registered OU to another, but old AWS Config rules remain. The account is in an inconsistent state.

Action to take:

If the account move was intended:

- Terminate the account in Service Catalog.
- Enroll it again.
- *Context/impact:* Deployed AWS Config rules don't match the configuration dictated by the destination OU.
- AWS Config rules may remain from the previous OU, causing unintended spending.
- Attempts to re-enroll or update the account will fail due to resource naming conflicts.

If the account move was unintended:

- Return the account to its original OU.
- Update the account from Service Catalog.
- In the launch parameters, enter the OU that the account was originally in.
- *Context/impact:* If the account is not returned to its original OU, its state will be inconsistent with the controls dictated by the new OU it's in.
- Updating an account is not a valid remediation, because it does not delete the AWS Config rules associated with its previous OU.

Failure Error that Mentions AWS Config

If AWS Config is enabled in any AWS Region supported by AWS Control Tower, you may receive an error message because a pre-check has failed. The message might not seem to explain the problem adequately, due to some underlying behavior of AWS Config.

You may receive an error message, similar to one of these:

- `AWS Control Tower cannot create an AWS Config delivery channel because one already exists. To continue, delete the existing delivery channel and try again`
  .
- `AWS Control Tower cannot create an AWS Config configuration recorder because one already exists. To continue, delete the existing delivery channel and try again`
  .

Common cause: When the AWS Config service is enabled on an AWS account, it creates a configuration recorder and delivery channel with a default naming. If you disable the AWS Config service through the console, it does not delete the configuration recorder or the delivery channel. You must delete them through the CLI, or modify them for AWS Control Tower use. If the AWS Config service is enabled in any one of the Regions supported by AWS Control Tower, it can result in this failure.

If the account has existing AWS Config resources, see Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.

Action to take: Delete the configuration recorder and delivery channel in all supported regions. Disabling AWS Config is not enough, the configuration recorder and delivery channel must be deleted by means of the CLI. After you've deleted the configuration recorder and delivery channel from the CLI, you can try again to launch AWS Control Tower and enroll the account.

If you are in the process of deploying a provisioned product, you must delete the provisioned product before you retry. Otherwise, you may see an error message similar to this one:

- `An error occurred` **`(InvalidParametersException)`** `when calling the` **`ProvisionProduct`** `operation: A stack named` *`Stackname`* `already exists.`

In the message, *`Stackname`* specifies the name of the stack.

Here are some example AWS Config CLI commands you can use to determine the status of your configuration recorder and delivery channel.

View commands:

- `aws configservice describe-delivery-channels`
- `aws configservice describe-delivery-channel-status`
- `aws configservice describe-configuration-recorders`
- `The normal response is something like "name": "default"`

Delete commands:

- `aws configservice stop-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
- `aws configservice delete-delivery-channel --delivery-channel-name` *`NAME-FROM-DESCRIBE-OUTPUT`*
- `aws configservice delete-configuration-recorder --configuration-recorder-name` *`NAME-FROM-DESCRIBE-OUTPUT`*

For more information, see the AWS Config documentation

- Managing the Configuration Recorder (AWS CLI)
- Managing the Delivery Channel

No Launch Paths Found Error

When you're trying to create a new account, you may see an error message similar to this one:

```
No launch paths found for resource: prod-dpqqfywxxxx
```

This error message is generated by AWS Service Catalog, which is the integrated service that helps provision accounts in AWS Control Tower.

Common Causes:

- You may be logged in as root. AWS Control Tower does not support creating accounts when you're logged in as root user.
- Your IAM Identity Center user has not been added to the appropriate permission group. You may need to add your IAM Identity Center user to one of these permission groups: **AWSAccountFactory** (for end-user access) or **AWSServiceCatalogAdmins** (for admin access).
- If you are authenticated as an IAM user, you must add it to the AWS Service Catalog portfolio so that it has the correct permissions.
- This issue also occurs if you have the correct permissions, but AWS Control Tower drift is detected, and a drift repair is necessary. To repair most types of drift, choose **Reset** on the **Landing zone settings** page.

Received an Insufficient Permissions Error

It's possible that your account may not have the necessary permissions to perform certain work in certain AWS Organizations. If you encounter the following type of error, check all the permissions areas, such as IAM or IAM Identity Center permissions, to make sure your permission is not being denied from those places:

```
You have insufficient permissions to perform AWS Organizations API actions.
```

If you believe your work requires the action you're attempting, and you can't locate any relevant restriction, contact your system administrator or AWS Support.

Detective controls are not taking effect on accounts

If you've recently expanded your AWS Control Tower deployment into a new AWS Region, newly-applied detective controls do not take effect on new accounts you create **in any Region** until the individual accounts within OUs governed by AWS Control Tower are updated. Existing detective controls on existing accounts are still in effect.

If you try to enable a detective control before updating your accounts, you may see an error message similar to this one:

```
AWS Control Tower can't enable the selected control on this OU. AWS Control
Tower cannot apply the control on the OU ou-xxx-xxxxxxxx, because child
accounts have dependencies that are missing. Update all child accounts
under the OU, then try again.
```

Action to take: Update accounts.

To update your accounts from the AWS Control Tower console, see When to update AWS Control Tower OUs and accounts.

To update multiple individual accounts programmatically, you can use the APIs from AWS Service Catalog and the AWS CLI to automate the updates. For more information about how to approach the update process, see this Video Walkthrough. You can substitute the **UpdateProvisionedProduct** API for the **ProvisionProduct** API shown in the video.

If you have further difficulties with enabling detective controls on your accounts, contact AWS Support.

Rate exceeded error returned by the AWS Organizations API

Possible cause

Your workload was running while AWS Control Tower was running a daily scan to check whether your SCPs have drifted.

Steps to follow

If you encounter an API throttling or `rate exceeded` error, try these steps:

- Run your workloads at a different time. (Refer to the AWS Control Tower SCP invariance scan schedule by Region to find out when AWS Control Tower runs its audit scans.)
- If you are calling the APIs directly through HTTP: Use the AWS SDK, which automatically retries failed actions
- Request a limit increase through Service Quotas and AWS Support

An example of troubleshooting instructions for API throttling in Elastic Beanstalk can be found here: `https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-api-throttling-errors/`

Failure to move an Account Factory account directly from one AWS Control Tower landing zone to another AWS Control Tower landing zone

> **Warning**
>
> This practice does not meet the prerequisite for eligible account enrollment, because eligible accounts must be part of the same overall AWS Organization, and each organization may have only one landing zone. If you have tried to do this action and you find yourself receiving multiple error messages, here is some information that might be helpful.

To move an account that you've provisioned through Account Factory into another landing zone that's managed by AWS Control Tower, under another management account, you must remove all of the IAM roles and the stacks associated with that account from the original OU. Remove these resources from every Region in which the account is deployed.

> **Note**
>
> The best way to remove the resources is to deprovision the account in its original OU before you try to move it.

If you don't remove the resources, enrollment into the new OU will fail, somewhat spectacularly. You may encounter one or more error messages, and you will keep receiving similar error messages until the remaining roles and stacks are removed from every Region in which the account was deployed.

Each time you receive an error message, you must remove the account from the new OU, delete the old resource that is the subject of the error message, and then attempt to move the account back into the new OU. This process of removing-and-deleting must be repeated for every remaining resource, for every Region in which the account was deployed, possibly 10 or 20 times. These repeated errors occur because the account was provisioned into an OU with an SCP that prevents IAM role deletion. You can make the recovery process shorter by deleting all the account's resources before you retry.

The examples below represent the types of failure messages you may receive if undeleted roles and stacks remain. You would most likely see one of these messages at a time, for each time you attempt to enroll the account, as long as old resources remain.

The values of the resource ID strings have been modified for the examples. Their values will not be the same in an error message you may receive. You may see a message similar to the following examples:

- `AWS Control Tower cannot create the IAM role` *`aws-controltower-AdministratorExecutionRole`* `because the role already exists. To continue, delete the existing IAM role and try again.`
- `AWS Control Tower cannot create the IAM role` *`aws-controltower-ConfigRecorderRole`* `because the role already exists. To continue, delete the existing IAM role and try again.`
- `AWS Control Tower cannot create the IAM role` *`aws-controltower-ForwardSnsNotificationRole`* `because the role already exists. To continue, delete the existing IAM role and try again.`

Or you may see an error message about a stack set failure, similar to this one:

```
"Error\":\"StackSetFailState\",
\"Cause\":\"StackSetOperation on AWSControlTowerBP-BASELINE-CLOUDWATCH
with id 8aXXXXf5-e0XX-4XXa-bc4XX-dXXXXXee31
has reached SUCCEEDED state but has 1 NON-CURRENT stack instances;
here is the summary :{ StackSet Id:
AWSControlTowerBP-BASELINE-CLOUDWATCH:40XXXbf2-Xead-46a1-XXXa-eXXXXecb2ee2,
Stack instance Id:
arn:aws:cloudformation:eu-west-1:1X23456789XX:
            stack/StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-4feXXXXXX-ecXX-XXc6-
bXXX-4ae678/4feXXXXXX-ecX-4ae123458,
Status: OUTDATED,
Status Reason: ResourceLogicalId:ForwardSnsNotification,
ResourceType:AWS::Lambda::Function,
ResourceStatusReason:aws-controltower-NotificationForwarder already exists in stack
arn:aws:cloudformation:eu-west-1:1X23456789XX:
            stack/StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-4feXXXXXX-ecXX-XXc6-
bXXX-4ae678/4feXXXXXX-ecX-4ae123458.
```

After all of the remaining resources are removed from the first OU, you'll be able to invite, provision, or enroll the account into the new OU successfully.

AWS Support

If you want to move your existing member accounts into a different support plan, you can sign in to each account with root account credentials, compare plans, and set the support level that you prefer.

We recommend that you update the MFA and account security contacts when you make changes to your support plan.

Types of baselines

A *baseline* in AWS Control Tower is a group of resources and specific configurations that you can apply to a target. The most common baseline target may be an organizational unit (OU). For example, you can enable a baseline with an OU selected as a target, to register that OU into AWS Control Tower.

During landing zone setup, the baseline target may be a shared account or the landing zone as a whole. Certain baselines may be enabled and updated based on your landing zone settings and configurations. AWS Control Tower creates and deploys the resources to the target in the way that the baseline specifies.

When you enable a baseline for a target, the baseline is represented as an AWS CloudFormation resource, called an `EnabledBaseline` resource.

AWS Control Tower includes two general types of baselines:

- Baseline types that can apply to an OU that's registered with AWS Control Tower, or to an OU that you intend to register by applying the baseline.
- Baseline types that can apply to a landing zone or shared account, during initial set up or during a landing zone update.

Baseline types that apply at the OU level, for registering and updating OUs

- **Name**: `AWSControlTowerBaseline`

  Description: Sets up resources and mandatory controls for member accounts within the target OU, required for AWS Control Tower governance.

  Consideration: This baseline retains the settings of the landing zone **Region deny** control. In other words, if a Region is not allowed at the landing zone level, that Region is not allowed for that OU when you call the `EnableBaseline` API to register an OU.

> **Note**
>
> The OU-level Region deny control has no way to allow Regions that the landing zone Region deny control does not allow.

For more information, see How SCPs work with deny in the AWS Organizations documentation.

Recommendation: We recommend that you confirm the Regions in which your target OU may be running workloads, and check the results against the landing zone Region deny control, before you call the `EnableBaseline` API for the OU, or you could lose access to resources in certain Regions.

- **Name**: `BackupBaseline`

  Description: This baseline sets up resources and controls for member accounts within the target OU. These are required so that integration with AWS Backup can automate your data backup across AWS services, and centralize your backup policy management.

  Consideration: Before you enable the `BackupBaseline` on a target OU, make sure that the `AWSControlTowerBaseline` is enabled on the target OU. That is, the target OU must be registered in AWS Control Tower.

  - You can choose to activate AWS Backup during the process of creating your AWS Control Tower landing zone, or during a landing zone update process.
  - The `BackupBaseline` is compatible with landing zone versions 3.1 and later.
  - The `BackupBaseline` is not applied to the management account.

> **Note**
>
> Landing zone baselines behave differently than OU-level baselines.

Baseline types that may apply to your landing zone or shared accounts

AWS Control Tower enables the baselines that apply at the landing zone level automatically, as part of the landing zone setup and update process. Baselines for your landing zone may change as you change your landing zone settings. For example, if you opt in for IAM Identity Center, AWS Control Tower can enable the latest version of the `IdentityCenterBaseline` baseline on your landing zone.

You can view the enabled baselines for your landing zone with the `ListEnabledBaselines` API call.

> **Note**
>
> Only the `AWSControlTowerBaseline` can be applied directly with the `EnableBaseline` API. Other baselines are managed automatically (`AuditBaseline`, `LogArchiveBaseline`). The status of `IdentityCenterBaseline` is provided as information when you apply the `AWSControlTowerBaseline`.

- **Name**: `AuditBaseline`

  Description: Sets up resources to monitor security and compliance of accounts in your organization. You cannot change this baseline, it is deployed by AWS Control Tower.
- **Name**: `LogArchiveBaseline`

  Description: Sets up a central repository for logs of API activities and resource configurations from accounts in your organization. You cannot change this baseline, it is deployed by AWS Control Tower.
- **Name**: `IdentityCenterBaseline`

  Description: Sets up shared resources for IAM Identity Center, which prepares the `AWSControlTowerBaseline` to set up Identity Center access for accounts.

  Consideration: This baseline works only when you've selected IAM Identity Center as your identity provider at the time you set up your landing zone initially, or if you subsequently change your landing zone settings to enable IAM Identity Center for your landing zone. If you're using a different identity provider, you won't have access to enable this baseline.

- **Name**: `BackupCentralVaultBaseline`

  Description: Sets up the central AWS Backup vault in your organization.
- **Name**: `BackupAdminBaseline`

  Description: Sets up delegated admin and the AWS Backup Audit Manager.

Partial enrollment of accounts

When you're working with baselines, an account can be placed into a state called **Partially enrolled**.

This state can occur if you re-register an OU by calling the `ResetEnabledBaseline` API, because AWS Control Tower applies only the mandatory resources to the accounts in the target OU. An account that is missing the optional resources (controls) for its parent OU is marked as **Partially enrolled**.

If you move an unenrolled account into a registered OU and then call the `ResetEnabledBaseline` API on the OU to enroll that account, AWS Control Tower applies the resources associated with the `AWSControlTowerBaseline` to the newly-enrolled account. However, optional controls enabled for this OU are not applied to the account. The account remains in a **Partially enrolled** state.

To enroll the account fully, choose **Re-register** or **Update account** in the console. When you select these operations from the console, AWS Control Tower applies all of the resources of that OU to the newly-enrolled account, including the optional controls that are activated for that OU.

Variation in operations between the AWS Control Tower console and APIs for baselines

When you change the governance status of an OU, the AWS Control Tower console performs more operations for you automatically, compared to changing governance by means of the APIs for baselines.

Differences

- **Registering and provisioned products**

When you register an OU through the console, AWS Control Tower creates Service Catalog products for the OU's member accounts, as part of enrolling each account. When you register an OU by means of the `EnableBaseline` API and the `AWSControlTowerBaseline`, AWS Control Tower does not create provisioned products for the member accounts in the OU.

- **Deregister an OU**

  Any time you deregister an OU, you must first remove all member accounts and nested OUs. Then, AWS Control Tower removes all controls that are applied to the OU.

  - If you select **Delete OU** the OU from the console, AWS Control Tower proceeds to deregister and then delete the OU from your organization.
  - However, if you deregister the OU by calling the `DisableBaseline` API to remove the `AWSControlTowerBaseline` from the OU, AWS Control Tower does not delete the OU from your organization, the OU is still present in the organization, unregistered.

Enabled baselines and member accounts

When you enable a baseline on an OU, that configuration is inherited by the OU's member accounts. Due to the fact of inheritance, we call it a *child enabled baseline* when we refer to the account. The baseline that's applied to the OU is called the *parent enabled baseline*. The parent enabled baseline controls the configuration of its child enabled baselines. It is similar to how a control, when enabled on an OU, applies to every account within the OU.

View an account's baseline status

AWS Control Tower does not allow you to target accounts directly with baselines. However, you can track the enablement and drift status of each member account by means of their inherited child enabled baselines. To view the status of your accounts, you can call the `ListEnabledBaselines` API with the `includeChildren` feature flag.

Baselines and versioning defaults

If your AWS Control Tower landing zone is already set up, and then you choose to enable a landing zone baseline, AWS Control Tower enables the latest version of the baseline that is compatible with your landing zone version. If you choose to enable a baseline for an OU that is not already registered with AWS Control Tower, AWS Control Tower provides the latest compatible version of the baseline for that OU, automatically.

Compatibility of OU baselines and landing zone versions

AWS Control Tower baselines allow you to set a governance standard at the OU level, rather than at the landing zone level, if your business requires it. The baseline called `AWSControlTowerBaseline` is available to help register your OUs with AWS Control Tower.

> **Note**
>
> A *baseline* is a group of controls and resources that work together to establish a stable governance environment within your landing zone.

When you enable a baseline on an OU, by callling the `EnableBaseline` API in AWS Control Tower, you must specify a baseline version that's compatible with your current AWS Control Tower landing zone version. After you specify a baseline, all member accounts in an OU follow the baseline given for the OU. In other words, new accounts are provisioned with the updated baseline, and existing member accounts become governed according to the new baseline.

If you do not select a baseline for your existing OUs and accounts, the landing zone version determines the entire governance posture, by default. However, each registered OU in your landing zone is assigned a baseline version, which is the latest baseline compatible with your current landing zone version. Therefore, each OU and enrolled member account has an associated baseline, even if you never assign a baseline specifically.

For the OU-level baseline, `AWSControlTowerBaseline`, the table that follows shows the compatibility of baselines with AWS Control Tower landing zone versions.

Baseline version	Landing zone versions	Included blueprints	Included controls	Change from previous baseline
1.0	2.0 to 2.7	BP_BASELINE_CLOUDTRAIL, BP_BASELINE_CLOUDWATCH, BP_BASELI	All mandatory controls	None

Baseline version	Landing zone versions	Included blueprints	Included controls	Change from previous baseline
		NE_CONFIG, BP_BASELINE_ROLES, BP_BASELINE_SERVICE_ROLES, IAM Resources		
2.0	2.8 to 2.9	BP_BASELINE_CLOUDTRAIL, BP_BASELINE_CLOUDWATCH, BP_BASELINE_CONFIG, BP_BASELINE_ROLES, BP_BASELINE_SERVICE_ROLES, Config SLR, IAM resources	All mandatory controls	Added AWS Config service-linked role (SLR) and new Config blueprint to use the SLR

Baseline version	Landing zone versions	Included blueprints	Included controls	Change from previous baseline
3.0	3.0 to 3.1	BP_BASELINE_CLOUDWATCH, BP_BASELINE_CONFIG, BP_BASELINE_ROLES, BP_BASELINE_SERVICE_ROLES, Config SLR, IAM resources	All mandatory controls	New AWS Config blueprint. Change to record global resources only in home Region. Removed CloudTrail blueprint
4.0	3.2 to 3.3	BP_BASELINE_CLOUDWATCH, BP_BASELINE_CONFIG, BP_BASELINE_ROLES, BP_BASELINE_SERVICE_LINKED_ROLE, BP_BASELINE_SERVICE_ROLES, Config SLR, IAM resources	All mandatory controls	New SLR blueprint

For more information about specific resources created in accounts when you set up your landing zone, see Resources created in the shared accounts.

If you update your landing zone to a version that supports a newer `AWSControlTowerBaseline` baseline version, and the new landing zone version is compatible with your existing baseline version, your OU state changes to **Update available**.

- You can continue to use account factory and other features without updating the OU baseline immediately, except in the case of a landing zone update from 2.x to 3.x.
- New accounts enrolled in this OU receive resources based on the existing baseline version until the baseline version is updated (with the **Extend governance** feature in the console, or by means of the `UpdateEnabledBaseline` API).
- After you update the baseline version, all accounts within that OU receive resources based on the new baseline version.

> **Note**
>
> If you update your AWS Control Tower landing zone from any version 2.X to any version 3.X, you also must update the baseline version on your OUs, due to the change from account-level to organization-level AWS CloudTrail trails. In the console, your OU will show a status of **Update required**.

Considerations for baselines

- If your OU requires a baseline update, you cannot provision new accounts or enroll existing accounts into that OU.
- After a landing zone update, if you also plan to update an OU baseline, you must re-register the OU or update your OU baseline version programmatically.
- We recommend that you update to the highest compatible baseline for the landing zone version you're using, so that you gain all the benefits of the landing zone and the baseline combined. For example, if you update to landing zone version 3.3, you can keep using baseline 3.0, but you do not get every benefit of landing zone version 3.3 unless you also update to baseline 4.0.
- Baseline updates cannot be rolled back.

- Baseline enablement targets one OU at a time. Therefore, nested OUs are not updated automatically when the parent OU is updated. We recommend that you update the parent OU before you update the nested OUs.
- When you call the `UpdateEnabledBaseline` API or re-register an OU from the console, the OU retains all controls that were enabled before the baseline update.
- When multiple baseline versions are compatible with your landing zone version, you must use the latest baseline version if you enable a baseline on an unmanaged OU.

Examples: Register an AWS Control Tower OU with APIs only

This walkthrough of examples is a companion document. For explanations, caveats, and more information, see Types of baselines.

Prerequisites

You must have an existing OU that is not registered with AWS Control Tower, and which you would like to register. Or, you must have a registered OU that you would like to re-register for purposes of updating.

Register an OU

1. Check whether the `IdentityCenterBaseline` is enabled for the landing zone. If so, get the Identity Center Enabled Baseline identifier.

```
aws controltower list-baselines --query 'baselines[?name==`IdentityCenterBaseline`].
[arn]'
```

```
aws controltower list-enabled-baselines --query 'enabledBaselines[?
baselineIdentifier==`<Identity Center Baseline Arn>`].[arn]'
```

2. Get the ARN of the target OU.

```
aws organizations describe-organizational-unit --organizational-unit-id
 <Organizational Unit ID> --query 'OrganizationalUnit.[Arn]'
```

3. Get the ARN of the `AWSControlTowerBaseline` baseline.

```
aws controltower list-baselines --query 'baselines[?name==`AWSControlTowerBaseline`].
[arn]'
```

4. Create the `AWSControlTowerBaseline` baseline on the target OU.

   If the Identity Center Baseline is enabled:

```
aws controltower enable-baseline --baseline-identifier <AWSControlTowerBaseline ARN>
 --baseline-version <BASELINE VERSION> --target-identifier <OU ARN> --parameters
 '[{"key":"IdentityCenterEnabledBaselineArn","value":"<Identity Center Enabled
 Baseline ARN>"}]'
```

   If the Identity Center Baseline is not enabled, omit the `parameters` flag, as follows:

```

aws controltower enable-baseline --baseline-identifier <AWSControlTowerBaseline ARN>
 --baseline-version <BASELINE VERSION> --target-identifier <OU ARN>
```

Re-register an OU

After you make updates to landing zone settings, or update your landing zone version, you must **Re-register** OUs to give them the latest changes. Follow these steps to re-register an OU programmatically, by resetting the associated `EnabledBaseline` resource.

1. Get the ARN of the target OU to re-register.

```
aws organizations describe-organizational-unit --organizational-unit-id <OU ID> --
query 'OrganizationalUnit.[Arn]'
```

2. Get the ARN of the `EnabledBaseline` resource for the target OU.

```
aws controltower list-enabled-baselines --query 'enabledBaselines[?
targetIdentifier==`<OUARN>`].[arn]'
```

3. Reset the Enabled Baseline.

```
aws controltower reset-enabled-baseline --enabled-baseline-
identifier <EnabledBaselineArn>
```

Examples for baseline API usage

This section contains examples of input and output parameters for the AWS Control Tower baseline APIs.

DisableBaseline

For more information about this API operation, see DisableBaseline.

`DisableBaseline` input:

```
{
    "enabledBaselineIdentifier": "arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/AB12CD34EF56GH789"
}
```

`DisableBaseline` output:

```
{
    "operationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f"
}
```

`DisableBaseline` CLI example:

```
aws controltower disable-baseline \
    --enabled-baseline-identifier arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/AB12CD34EF56GH789 \
    --region us-west-2
```

EnableBaseline

For more information about this API operation, see EnableBaseline.

`EnableBaseline` input:

```
{
    "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline:17BSJV3IGJ2QSGA2",
    "targetIdentifier": "arn:aws:organizations::123456789012:ou/o-kgj0txdhpa/ou-
r9mj-4j3mzjql",
    "baselineVersion": "3.0",
    "parameters": [
```

```
        {
            "key": "IdentityCenterEnabledBaselineArn",
            "value": "arn:aws:controltower:us-west-2:123456789012:enabledbaseline/
XAHCR4CJTSI4W07MZ"
        }
    ]
}
```

EnableBaseline output, returning a new resource:

```
{
    "operationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f",
    "arn": "arn:aws:controltower:us-west-2:123456789012:enabledbaseline/
XAGF7TNOHRD7ES5VV"
}
```

EnableBaseline CLI example:

This example shows enabling a baseline for an AWS Organizations organization that has the landing zone opted-in to AWS IAM Identity Center access, managed by AWS Control Tower. To retrieve your Identity Center EnabledBaseline identifier, you can call the ListEnabledBaselines API, filtering on the Identity Center baseline: (arn:aws:controltower:*Region*::baseline/LN25R72TTG6IGPTQ)

```
aws controltower list-enabled-baselines \
    --filter baselineIdentifiers=arn:aws:controltower:us-west-2::baseline/
LN25R72TTG6IGPTQ \
    --region us-west-2
```

The response will show the EnabledBaseline detail, which shows its identifier.

```
{
    "enabledBaselines": [
        {
            "arn": "arn:aws:controltower:us-west-2:123456789012:enabledbaseline/
XAHXS7P6C4I453EZC",
            "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline/
LN25R72TTG6IGPTQ",
            "targetIdentifier": "arn:aws:organizations::123456789012:account/o-
aq21sw43de5/123456789012",
            "statusSummary": {
                "status": "SUCCEEDED"
```

```
            }
        }
    ]
}
```

> **Note**
>
> Make note of the ARN value from the response, and pass this value as a parameter to enable the default baseline.

```
aws controltower enable-baseline \
    --baseline-identifier arn:aws:controltower:us-west-2::baseline/17BSJV3IGJ2QSGA2 \
    --baseline-version 3.0 \
    --target-identifier arn:aws:organizations::123456789012:ou/o-aq21sw43de5/ou-po90-
lk87jh65 \
    --parameters
 '[{"key":"IdentityCenterEnabledBaselineArn","value":"arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHXS7P6C4I453EZC"}]' \
    --region us-west-2
```

For an organization with the landing zone opted-out from AWS Control Tower management of IAM Identity Center, enable the baseline without the parameter.

```
aws controltower enable-baseline \
    --baseline-identifier arn:aws:controltower:us-west-2::baseline/17BSJV3IGJ2QSGA2 \
    --baseline-version 3.0 \
    --target-identifier arn:aws:organizations::123456789012:ou/o-aq21sw43de5/ou-po90-
lk87jh65 \
    --region us-west-2
```

GetBaseline

For more information about this API operation, see GetBaseline.

`GetBaseline` input:

```
{
    "baselineIdentifier": "arn:aws:controltower:us-west-2::baseline/17BSJV3IGJ2QSGA2"
}
```

GetBaseline output:

```
{
    "arn": "arn:aws:controltower:us-west-2::baseline/17BSJV3IGJ2QSGA2",
    "name": "AWSControlTowerBaseline",
    "description": "Sets up resources and mandatory controls for member accounts within
 the target OU, required for AWS Control Tower governance.",
}
```

GetBaseline CLI example:

```
aws controltower get-baseline \
    --baseline-identifier arn:aws:controltower:us-west-2::baseline/17BSJV3IGJ2QSGA2 \
    --region us-west-2
```

GetBaselineOperation

For more information about this API operation, see GetBaselineOperation.

GetBaselineOperation input:

```
{
    "operationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f"
}
```

GetBaselineOperation output:

```
{
    "baselineOperation": {
        "operationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f",
        "operationType": "DISABLE_BASELINE",
        "status": "FAILED",
        "startTime": "2023-01-12T19:05:00Z",
        "endTime": "2023-01-12T19:45:00Z",
        "statusMessage": "Can't perform DisableBaseline on a parent target with
 governed child OUs"
    }
}
```

GetBaselineOperation CLI example:

```
aws controltower get-baseline-operation \
```

```
    --operation-identifier 58f12232-26be-4735-a3e9-dd30d90f021f \
    --region us-west-2
```

GetEnabledBaseline

For more information about this API operation, see GetEnabledBaseline.

`GetEnabledBaseline` input:


```
{
    "enabledBaselineIdentifier": "arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHCR4CJTSI4W07MZ"
}
```


`GetEnabledBaseline` output:


```
{
    "enabledBaselineDetails": {
        "arn": "arn:aws:controltower:us-west-2:123456789012:enabledbaseline/
XAHCR4CJTSI4W07MZ",
        "baselineIdentifier": "arn:aws:controltower:us-
west-2::baseline:17BSJV3IGJ2QSGA2",
        "baselineVersion": "3.0",
        "targetIdentifier": "arn:aws:organizations::123456789012:ou/o-kgj0txdhpa/ou-
r9mj-4j3mzjql",
        "statusSummary": {
            "status": "SUCCEEDED",
            "lastOperationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f"
        },
        "parameters": [
            {
                "key": "IdentityCenterEnabledBaselineArn",
                "value": "arn:aws:controltower:us-west-2:123456789012:enabledbaseline/
XAHCR4CJTSI4W07MZ"
            }
        ]
    }
}
```


`GetEnabledBaseline` CLI example:

```
aws controltower get-enabled-baseline \
```

```
    --enabled-baseline-identifier arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHXS7P6C4I453EZC \
    --region us-west-2
```

ListBaselines

For more information about this API operation, see ListBaselines.

`ListBaselines` input (using optional inputs):

```
{
    "nextToken": "AbCd1234",
    "maxResults": "4"
}
```

`ListBaselines` output:

```
{
    "baselines": [
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/4T4HA1KMO10S6311",
            "name": "AuditBaseline",
            "description": "Sets up resources to monitor security and compliance of
 accounts in your organization."
        },
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/J8HX46AHS5MIKQPD",
            "name": "LogArchiveBaseline",
            "description": "Sets up a central repository for logs of API activities and
 resource configurations from accounts in your organization."
        },
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/LN25R72TTG6IGPTQ",
            "name": "IdentityCenterBaseline",
            "description": "Sets up shared resources for AWS Identity Center, which
 prepares the AWSControlTowerBaseline to set up Identity Center access for accounts."
        },
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/17BSJV3IGJ2QSGA2",
            "name": "AWSControlTowerBaseline",
            "description": "Sets up resources and mandatory controls for member
 accounts within the target OU, required for AWS Control Tower governance."
        },
```

```
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/3WPD0NA6TJ9AOMU2",
            "name": "BackupCentralVaultBaseline",
            "description": "Sets up central AWS Backup vault in your organization."
        },
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/H6C5JFCJJ3CPU3J5",
            "name": "BackupManagerBaseline",
            "description": "Sets up delegated admin and AWS Backup Audit Manager."
        },
        {
            "arn": "arn:aws:controltower:us-east-1::baseline/APO9ATVPBKFRRGLK",
            "name": "BackupBaseline",
            "description": "Sets up local Backup vault and attach Backup policy."
        }
    ]
}
```


`ListBaselines` CLI example:

```
aws controltower list-baselines \
    --region us-west-2
```

ListEnabledBaselines

The `ListEnabledBaselines` API has an optional parameter that allows you to view the baselines as they apply to the accounts that are members of an OU. The examples that follow show some CLI commands you can use to view the baselines for an account. AWS Control Tower refers to these baselines, which are enabled on the OU, but apply to each account within the OU, as *child enabled baselines*, because they derive their governance configuration from the baselines that are applied on the OU.

For more information about this API operation, see ListEnabledBaselines.

`ListEnabledBaselines` input to show child enabled baselines:

```
aws controltower list-enabled-baselines --include-children
```

`ListEnabledBaselines` output to view child enabled baselines:


```
{
```

```
    "enabledBaselines": [
        {
            "arn": "arn:aws:controltower:us-east-1:666355521292:enabledbaseline/
XO2UQ1PC6BB5085S5",
            "baselineIdentifier": "arn:aws:controltower:us-east-1::baseline/
APO9ATVPBKFRRGLK",
            "baselineVersion": "1.0",
            "statusSummary": {
                "lastOperationIdentifier": "07d6d2b8-e357-4f96-ba00-98ea88143445",
                "status": "SUCCEEDED"
            },
            "targetIdentifier": "arn:aws:organizations::666355521292:ou/o-vaex10vaey/
ou-k86y-ld9k8vpu"
        },
        {
            "arn": "arn:aws:controltower:us-east-1:666355521292:enabledbaseline/
XAFPKQQXOJB50ZWQH",
            "baselineIdentifier": "arn:aws:controltower:us-east-1::baseline/
APO9ATVPBKFRRGLK",
            "baselineVersion": "1.0",
            "parentIdentifier": "arn:aws:controltower:us-
east-1:666355521292:enabledbaseline/XOIZ4G08CWB50ZWON",
            "statusSummary": {
                "lastOperationIdentifier": "3508793e-48c8-4895-965b-3dc6abd52b6b",
                "status": "SUCCEEDED"
            },
            "targetIdentifier": "arn:aws:organizations::666355521292:account/o-
vaex10vaey/183295447314"
        }
]
```

Note

In the previous example, the `parentIdentifier` field shows the enabled baseline of the parent OU for this child enabled baseline.

View all baselines applied on a specific target (OU or account):

```
aws controltower list-enabled-baselines \
    --filter '{
        "targetIdentifiers": ["TARGET_ARN"]
```

```
    }
```

View all OUs that have a specific baseline:

```
aws controltower list-enabled-baselines \
    --filter '{
        "baselineIdentifiers": ["BASELINE_ARN"]
    }'
```

View all OUs and accounts that have a specific baseline:

```
aws controltower list-enabled-baselines \
    --filter '{
        "baselineIdentifiers": ["BASELINE_ARN"]
    }'  \
    --include-children
```

View all accounts in an OU that have Baseline B enabled:

```
### First fetch the enabled baseline record for Baseline B on the OU
 aws controltower list-enabled-baselines \
    --filter '{
        "targetIdentifiers": ["OU_TARGET_ARN"],
        "baselineIdentifiers": ["BASELINE_ARN_FOR_BASELINE_B"]
    }'

### Call ListEnabled baseline to fetch all accounts that have their parent as the
 enabled baseline record on the OU
aws controltower list-enabled-baselines \
    --filter '{
        "parentIdentifiers": ["ENABLED_BASELINE_ARN_FOR_OU"]
    }' \
    --include-children
```

More about child enabled baselines

- You can use `GetEnabledBaseline` API to view detailed information about a specific child enabled baseline
- You can use the `GetBaselineOperation` API to view an operation performed on the child enabled baseline

- You cannot call any write APIs, such as `EnableBaseline`, `UpdateEnabledBaseline`, `ResetEnabledBaseline` or `DisableBaseline`, on a child enabled baseline directly.
- Child enabled baseline resources can be modified by means of the AWS Control Tower service only, through operations that are performed on the parent OU, or by means of Account Factory.

Examples for using filters:

`ListEnabledBaselines` input (no filters):

```
{
    "nextToken": "bde7-XX0c6fXXXXXX",
    "maxResults": 5
}
```

`ListEnabledBaselines` input (`baselineIdentifiers` filter only):

```
{
    "filter": {
        "baselineIdentifiers": ['arn:aws:controltower:us-
east-1::baseline/17BSJV3IGJ2QSGA2', 'arn:aws:controltower:us-
east-1::baseline/12GZU8CKZKVMS2AW']
    },
    "nextToken": "bde7-XX0c6fXXXXXX",
    "maxResults": 5
}
```

`ListEnabledBaselines` input (`targetIdentifiers` filter only):

```
{
    "filter": {
        "targetIdentifiers": ['arn:aws:organizations::123456789012:ou/o-s9511vn103/ou-
xqj7-fex1u317', 'arn:aws:organizations::123456789012:ou/o-s9511vn103/ou-xqj7-11q6n2cf']
    },
    "nextToken": "bde7-XX0c6fXXXXXX",
    "maxResults": 2
}
```

`ListEnabledBaselines` input (`baselineIdentifiers` and `targetIdentifiers` filters):

```
{
    "filter": {
        "baselineIdentifiers": ['arn:aws:controltower:us-
east-1::baseline/17BSJV3IGJ2QSGA2']
        "targetIdentifiers": ['arn:aws:organizations::123456789012:ou/o-s9511vn103/ou-
xqj7-fex1u317']
    },
    "nextToken": "bde7-XX0c6fXXXXXX",
    "maxResults": 5
}
```

`ListEnabledBaselines` output:

```
{
    "enabledBaselines": [
        {
            "arn": "arn:aws:controltower:us-east-1:123456789012:enabledbaseline/
XAHCR4CJTSI4W07MZ",
            "baselineIdentifier": "arn:aws:controltower:us-
east-1::baseline:17BSJV3IGJ2QSGA2",
            "baselineVersion": "3.0",
            "targetIdentifier": "arn:aws:organizations::123456789012:ou/o-kgj0txdhpa/
ou-r9mj-4j3mzjql",
            "statusSummary": {
                "status": "SUCCEEDED",
                "lastOperationIdentifier": "58f12232-26be-4735-a3e9-dd30d90f021f"
            }
        },
        {
            "arn": "arn:aws:controltower:us-east-1:123456789012:enabledbaseline/
XAJ9NKW88AA4W9CLL",
            "baselineIdentifier": "arn:aws:controltower:us-
east-1::baseline:17BSJV3IGJ2QSGA2",
            "baselineVersion": "4.0",
            "targetIdentifier": "arn:aws:organizations::123456789012:ou/o-s9511vn103/
ou-xqj7-fex1u317",
            "statusSummary": {
            "status": "FAILED",
                "lastOperationIdentifier": "81e02df1-2b4d-48f0-838f-3833b93dcdc0"
            }
        }
    ],
    "nextToken": "e2bXXXXX6cab"
```

```
}
```

CLI example with one type of filter (`baselineIdentifiers` filter):

```
aws controltower list-enabled-baselines \
    --filter baselineIdentifiers=arn:aws:controltower:us-
west-2::baseline/17BSJV3IGJ2QSGA2,arn:aws:controltower:us-west-2::baseline/
LN25R72TTG6IGPTQ \
    --region us-west-2
```

CLI example using multiple filters (`baselineIdentifiers` and `targetIdentifiers` filters):

```
aws controltower list-enabled-baselines \
    --filter targetIdentifiers=arn:aws:organizations::123456789012:ou/o-
aq21sw43de5/ou-po90-lk87jh65,baselineIdentifiers=arn:aws:controltower:us-
west-2::baseline/17BSJV3IGJ2QSGA2 \
    --region us-west-2
```

`ResetEnabledBaseline`

For more information about this API operation, see ResetEnabledBaseline.

`ResetEnabledbaseline` input:

```
{
    "enabledBaselineIdentifier": "arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAJ9NKW88AA4W9CLL"
}
```

`ResetEnabledBaseline` output:

```
{
    "operationIdentifier": "81e02df1-2b4d-48f0-838f-3833b93dcdc0"
}
```

`ResetEnabledBaseline` CLI example:

```
aws controltower reset-enabled-baseline \
    --enabled-baseline-identifier arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHXS7P6C4I453EZC \
```

```
    --region us-west-2
```

UpdateEnabledBaseline

For more information about this API operation, see UpdateEnabledBaseline.

`UpdateEnabledBaseline` input:

```
{
    "enabledBaselineIdentifier": "arn:aws:controltower:us-
east-1:123456789012:enabledbaseline/XAJ9NKW88AA4W9CLL",
    "baselineVersion": "4.0",
    "parameters": [
        {
            "key": "IdentityCenterEnabledBaselineArn",
            "value": "arn:aws:controltower:us-east-1:123456789012:enabledbaseline/
XAHCR4CJTSI4W07MZ"
        }
    ]
}
```

`UpdateEnabledBaseline` output:

```
{
    "operationIdentifier": "81e02df1-2b4d-48f0-838f-3833b93dcdc0"
}
```

`UpdateEnabledBaseline` CLI example:

```
aws controltower update-enabled-baseline \
    --enabled-baseline-identifier arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHXS7P6C4I453EZC \
    --baseline-version 4.0
    --parameters
 '[{"key":"IdentityCenterEnabledBaselineArn","value":"arn:aws:controltower:us-
west-2:123456789012:enabledbaseline/XAHXS7P6C4I453EZC"}]' \
    --region us-west-2
```

Additional information and links

This topic includes links to relevant blog posts, technical documentation, and related information that can help you as you work with AWS Control Tower. The sources cover some common use cases and best practices for AWS Control Tower capabilities, and some additional enhancements.

Tutorials and labs

- AWS Control Tower lab – These labs provide a high-level overview of common tasks related to AWS Control Tower.
- On the AWS Control Tower dashboard, choose **Get personalized guidance** if you have a use case in mind but you're not sure where to start.
- Try visiting this AWS Cloud Operations YouTube playlist and search AWS Control Tower to find videos that explain more about how to use AWS Control Tower functionality.

Networking

Set up repeatable and manageable patterns for networks in AWS. Learn more about design, automation, and appliances that are commonly used by customers.

- AWS Quick Start VPC Architecture– This Quick Start guide provides a networking foundation based on AWS best practices for your AWS Cloud infrastructure. It builds an AWS Virtual Private Network environment with public and private subnets where you can launch AWS services and other resources.
- Self-service VPCs in AWS Control Tower using AWS Service Catalog– This blog post describes a way to set up Account Factory so you can provision accounts with customized VPCs.
- Implementing Serverless Transit Network Orchestrator (STNO) in AWS Control Tower – This blog post demonstrates how to automate network connectivity access across accounts. This blog is intended for AWS Control Tower administrators, or those responsible for managing networks within their AWS environment.

Security, identity, and logging

Extend your security posture, integrate with external or existing identity providers, and centralize logging systems.

Security

- Automating AWS Security Hub Alerts with AWS Control Tower lifecycle events – This blog post describes how to automate Security Hub enablement and configuration in an AWS Control Tower multi-account environment on existing and new accounts.
- Enabling AWS Identity and Access Management – This blog post describes how to enhance your organizational security visibility by enabling and centralizing IAM Access Analyzer findings.
- AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. You can use it to share configuration information in a secure location, for use by AWS Systems Manager and by AWS CloudFormation. For example, you can store a list of Regions in which you want to deploy conformance packs.

Identity

- Link Azure AD user identity into AWS accounts and applications for single sign-on – This blog post describes how to use Azure AD with IAM Identity Center and AWS Control Tower.
- Manage access to AWS centrally for Okta users with AWS IAM Identity Center – This blog post describes how to use Okta with IAM Identity Center and AWS Control Tower.

Logging

- AWS Centralized Logging Solution – This solutions post describes the Centralized Logging solution which enables organizations to collect, analyze, and display logs on AWS across multiple accounts and AWS Regions.
- For information about viewing your AWS Config resources, see the Config Resource Compliance Dashboard.

Deploying resources and managing workloads

Deploy and manage resources and workloads.

- Getting Started Library integration – This blog post describes Getting Started portfolios you can use.
- Continuous deployment of Cloud Custodian to AWS Control Tower

Working with existing organizations and accounts

Work with existing AWS organizations and accounts.

- Enroll an account – This user guide topic describes how to enroll an existing AWS account in AWS Control Tower.
- Bring an account under AWS Control Tower – This blog post describes how to deploy AWS Control Tower into your existing AWS organizations.
- Extend AWS Control Tower governance using AWS Config conformance packs – This blog post describes how to deploy AWS Config conformance packs to assist with bringing existing accounts and organizations into governance by AWS Control Tower.
- How to Detect and Mitigate Guardrail Violation with AWS Control Tower – This blog post describes how to add controls and how to subscribe to SNS notifications so that you can be notified by email of control compliance violations.

Automation and integration

Automate account creation and integrate lifecycle events with AWS Control Tower.

- Lifecycle events – This blog post describes how to use lifecycle events with AWS Control Tower.
- Automate account creation – This blog post describes how to set up automated account creation in AWS Control Tower.
- Amazon VPC flow log automation – This blog post describes how to automate and centralize Amazon VPC Flow Logs in a multi-account environment.
- Automate VPC tagging with AWS Control Tower lifecycle events– This blog post describes how to automate resource tagging for VPCs, by means of lifecycle events in AWS Control Tower.
- Automated account management – This blog post describes how to automate account management tasks after your AWS Control Tower environment is set up.

Migrating workloads

Use other AWS services with AWS Control Tower to assist in workload migration.

- CloudEndure migration – This blog post describes how to combine CloudEndure and other AWS services with AWS Control Tower to assist in workload migration.

Related AWS services

AWS Control Tower acts as an orchestration layer for AWS Organizations. Therefore, by means of the AWS Organizations console and APIs, you have access to over 20 other AWS services that work with AWS Control Tower. These additional services are not accessible directly through the AWS Control Tower console.

- For a full list of services available to AWS Control Tower by means of AWS Organizations, see [AWS services that you can use with AWS Organizations].
- To enable multi-account capabilities for these related AWS services, you must enable trusted access. For more information, see [Using AWS Organizations with other AWS services].

> **Note**
>
> Remember that AWS IAM Identity Center, AWS Config, and AWS CloudTrail are set up for you in AWS Control Tower and fully integrated. You do not need to modify your trusted access or delegated administration settings for these services.

- Some AWS services available through AWS Organizations can use delegated administration, including AWS Systems Manager and AWS Firewall Manager. For more information, see [Configuring a Delegated Administrator], and [Enabling a delegated administrator account for Firewall Manager]. Also see this video, [Set up security groups with AWS Firewall Manager].

AWS Marketplace solutions

Discover solutions from AWS Marketplace.

- [AWS Control Tower Marketplace] – AWS Marketplace offers a broad range of solutions for AWS Control Tower to help you integrate third-party software. These solutions help solve key infrastructure and operational use cases including identity management, security for a multi-account environment, centralized networking, operational intelligence, and security information and event management (SIEM).

AWS Control Tower release notes

The following sections show details about AWS Control Tower releases, since the inception of the service. Some releases require an update for an AWS Control Tower landing zone, and other releases are incorporated into the service automatically.

Features and releases are listed in reverse chronological order (most recent first) based on the date on which they were officially announced to the public.

Topics

January 2025 - Present

Since January 2025, AWS Control Tower has released the following updates:

- AWS Control Tower supports IPv6 addresses
- AWS Control Tower reduces drift

  Account Factory for Terraform version 1.15.0 available
- Updated controls with Nitro instance types
- AWS Control Tower available in Asia Pacific (Taipei) Region
- AWS Control Tower supports PrivateLink
- Support for additional industry frameworks, updated metadata
- Service-linked AWS Config controls
- Enabled controls console view gives centralized visibility
- the section called “Account Factory for Terraform (AFT) supports new configurations at deployment”

- AWS Control Tower introduces account-level reporting for baseline APIs
- the section called “AWS Control Tower available in AWS Asia Pacific (Thailand) and Mexico (Central) Regions”
- Additional AWS Config controls available
- Deregister and delete actions for OUs
- the section called “Control Catalog supports IPv6 addresses”

AWS Control Tower reduces drift

August 20, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has updated the functionality of service control policy (SCP) drift. With this release, two types of governance drift are handled directly by AWS Control Tower and no longer cause drift in your environment.

Two types of governance drift removed

- **SCP attached to managed OU** – This type of drift occurred when an SCP for a control was attached to any other OU. This occurrence was especially common when you updated OUs from outside of the AWS Control Tower console.
- **SCP attached to member account** – This type of drift commonly occured when an SCP for a control was attached to an account from outside the AWS Control Tower console.

For more information about drift, see Detect and resolve drift in AWS Control Tower.

AWS Control Tower supports IPv6 addresses

August 18, 2025

(No update required for AWS Control Tower landing zone.)

The AWS Control Tower API now supports Internet Protocol version 6 (IPv6) addresses through our new dual-stack endpoints. The existing endpoints supporting IPv4 remains available for backwards compatibility. The new dual-stack domains are available either from the internet or from within an Amazon Virtual Private Cloud (VPC) using AWS PrivateLink.

Account Factory for Terraform version 1.15.0 available

July 28, 2025

(No update required for AWS Control Tower landing zone.)

Version 1.15.0 of AWS Control Tower Account Factory for Terraform (AFT) is available. For more information, see the AFT GitHub repository.

Updated controls with Nitro instance types

July 24, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has updated the eight proactive controls in the Control Catalog (formerly called the Control Library) that enforce the Amazon EC2 intance types. This update allows you to instantiate some new Nitro instance types, and it removes some deprecated *u-series* instance types.

Updated controls

- `CT.AUTOSCALING.PR.10`
- `CT.AUTOSCALING.PR.11`
- `CT.EC2.PR.15`
- `CT.EC2.PR.16`
- `CT.EC2.PR.17`
- `CT.EC2.PR.18`
- `CT.EC2.PR.19`
- `CT.EC2.PR.20`

New instance types available

- `c8gd`
- `c8gn`
- `i7i`
- `m8gd`
- `p6-b200`

- `r8gd`

Instance types removed

- `u-12tb1`
- `u-18tb1`
- `u-24tb1`
- `u-9tb1`

The updated controls are available in all AWS Regions where AWS Control Tower is available. For a list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in Asia Pacific (Taipei) Region

July 23, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is now available in the Asia Pacific (Taipei) Region:

For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower supports PrivateLink

June 30, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports AWS PrivateLink. You can invoke AWS Control Tower and Control Catalog APIs from within your Amazon Virtual Private Cloud (VPC) without traversing the public internet. AWS PrivateLink provides private connectivity among virtual private clouds (VPCs), supported services and resources, and your on-premises networks. AWS PrivateLink support for AWS Control Tower is available in all AWS Regions where AWS Control Tower is available.

Support for additional industry frameworks, updated metadata

June 12, 2025

(No update required for AWS Control Tower landing zone.)

With this release, AWS Control Tower expands to include support for 10 industry frameworks. For a list of frameworks, see Frameworks supported.

For example, you can get started by navigating to the Control Catalog page in the AWS Control Tower console and searching for a framework, such as **PCI-DSS-v4.0**, to view all controls related to that framework. Or you can examine controls and frameworks programmatically, by calling the new `ListControlMappings` API.

The metadata definitions associated with controls are changing, to better support these additional industry framewoks. The changes to metadata can affect the way you evaluate controls for enablement. For example, the values for NIST, PCI, and CIS metadata may have changed. We recommend that you r*eview the mappings* for your enabled controls, on the **Control details** page in the console.

In the console and the API, we introduced 3 new metadata fields. Collectively, these fields describe a hierarchy that helps you understand how to categorize and enable controls. The fields are: **Domain**, **Objective**, and **Common control**. We have redefined our control objectives to align better with the broader scope of industry frameworks that are available. For more information about this hierarchy, see Ontology overview.

- These metadata changes are reflected in the AWS Control Tower console, and the console experience is consistent across the AWS Control Tower and AWS Config consoles.
- To view control information in the AWS Control Tower console, you must add additional `controlcatalog` permissions to your IAM policies. For more information, see Permissions required to use the AWS Control Tower console.
- Each control now has a new field called `GovernedResources`, which shows the resource types that the control governs. In some cases, this field shows the service prefix for the resources, and in other instances, it can be blank. For more information, see `GetControl` and `ListControls`.

With this release, we have renamed the *Controls Library* to *Control Catalog*, for consistency with other terminology.

Service-linked AWS Config controls

June 12, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower announces support for the AWS Control Tower detective controls to be deployed as service-linked AWS Config rules.

With this release, AWS Control Tower now deploys service-linked Config rules directly in your enrolled accounts, replacing the previous method of deployment with AWS CloudFormation stack sets. This change significantly improves deployment speed. Also, these service-linked Config rules help to ensure consistent governance of your resources, because they prevent unintentional configuration drift that could be caused by manual changes to AWS CloudFormation stack sets or Config rules.

Going forward, all AWS Control Tower controls implemented by AWS Config rules will be deployed with this mechanism, which directly calls the AWS Config APIs.

> **Important**
>
> Before you adopt service-linked Config rules, review the existing customizations, such as remediations, that you have made to Config rules outside of AWS Control Tower, because these customizations will be removed during the transition. The AWS Config APIs do not support adding remediation configurations for service-linked AWS Config rules. See `PutRemediationConfigurations`.

Details and action required

- When you **Update** or **Reset** your landing zone, AWS Control Tower updates the **Mandatory** controls that govern the Security OU. To complete the upgrade, you also must **Reset** each of the detective controls that are implemeneted with AWS Config rules, or **Re-register** the OU.
- The full scope of this upgrade applies to you if your AWS Control Tower landing zone version is 3.2 or above. When you apply this update, your existing AWS Config rules are changed to become service-managed Config rules, along with the new deployment method.
- If your landing zone is version 3.1 or below, any new Config rules will be deployed with the new method, no longer with Stack Sets. Your existing Config rules are NOT updated to become service-managed Config rules. They will remain of the standard type.
- You can identify service-linked config rules by their resource ARN, which has the form:

```
arn:aws:config:*:*:config-rule/aws-service-rule/controltower.*/*
```

The intended functionality of controls, when implemented by service-linked AWS Config rules, has not changed. The detective, service-linked Config rules in AWS Control Tower can identify non-compliant resources within your accounts, such as policy violations, and provide alerts through the dashboard. To maintain consistency, prevent configuration drift, and simplify your overall user experience, these rules now can be modified only through AWS Control Tower.

As part of this release, we added four new permissions to the policy for the service-linked role (SLR) `AWSServiceRoleForAWSControlTower`, so that you can enable and disable service-linked AWS Config rules for your enrolled accounts.

```
config:DescribeConfigRules
config:TagResource
config:PutConfigRule
config:DeleteConfigRule
```

Enabled controls console view gives centralized visibility

May 21, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower added a new page in the console that displays all of your enabled controls in a single, centralized view. Previously, controls were viewable only with the account or OU on which they were enabled. The consolidated view makes it easier for you to identify gaps in your control governance, at scale.

On the **Enabled controls** page, you can filter the controls according to behavior: Preventive, Detective, or Proactive. You also can filter according to control implementation, such as SCP. For each control, you can see how many OUs have this control enabled.

To see the **Enabled controls** page, navigate to the **Controls** section in the AWS Control Tower console.

Account Factory for Terraform (AFT) supports new configurations at deployment

May 13, 2025

(No update required for AWS Control Tower landing zone.)

The AWS Control Tower account customization framework, Account Factory for Terraform (AFT), now supports three additional optional configurations at the time of deployment. You can deploy AFT into a custom virtual private cloud (VPC), specify the Terraform project name for your AFT deployment, and tag the resources that AFT creates.

For more information, see [Deploy AWS Control Tower Account Factory for Terraform (AFT)]().

AWS Control Tower introduces account-level reporting for baseline APIs

May 12, 2025

(No update required for AWS Control Tower landing zone.)

You can now view *drift* and *account enrollment* statuses programmatically for your governed accounts, by calling the baseline APIs. With this capability, you can identify when account and OU baseline configurations are *drifted*, or out of sync. To view the drift status programmatically, you can call the `ListEnabledBaselines` API for your enabled baselines. To view statuses for individual accounts programmatically with the `ListEnabledBaselines` API, use the `includeChildren` flag. You can filter by these statuses, and see only the accounts and OUs that require your attention.

The `AWSControlTowerBaseline` sets up best practice configurations, controls, and resources that are required for governance. When you enable this baseline on an organizational unit (OU), member accounts within the OU are enrolled into AWS Control Tower automatically. The AWS Control Tower baseline APIs include AWS CloudFormation support, which allows you to build automations that manage your OUs and accounts with infrastructure as code (IaC).

To learn more about these APIs, review Baselines in the *AWS Control Tower User Guide*. The baseline APIs and newly launched reporting capabilities for *drift* and *account enrollment* status are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in AWS Asia Pacific (Thailand) and Mexico (Central) Regions

May 9, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is now available in the following AWS Regions:

- Asia Pacific (Thailand)
- Mexico (Central)

For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

Additional AWS Config controls available

April 11, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports an additional 223 managed AWS Config rules for various use cases, such as security, cost, durability, and operations. With this launch, you can now use AWS Control Tower to search and discover the AWS Config rules that you need to govern your multi-account environment; then enable and manage the controls directly from AWS Control Tower.

To get started from the AWS Control Tower console, go to the Control Catalog and search for controls with the implementation filter AWS Config. You can enable the controls directly from the AWS Control Tower console.

For more details, see Integrated AWS Config controls available in AWS Control Tower.

With this launch, we've updated the `ListControls` and `GetControl` APIs to support three new fields: **CreateTime**, **Severity**, and **Implementation**, which you can use when searching for a control in Control Catalog. For example, you can now programmatically find high-severity AWS Config rules that were created after your last evaluation.

You can search for the new AWS Config rules in all AWS Regions where AWS Control Tower is available. To deploy a rule, refer to the list of supported AWS Regions for that rule, to see where it can be enabled.

Deregister and delete actions for OUs

April 8, 2025

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports separate console actions to deregister an OU and to delete an OU. You must deregister the OU before you delete it. You can remove an OU from AWS Control Tower by deregistering it.

For more information, see Remove an OU.

Control Catalog supports IPv6 addresses

April 2, 2025

(No update required for AWS Control Tower landing zone.)

The AWS Control Tower Control Catalog API now supports Internet Protocol version 6 (IPv6) addresses through our new dual-stack endpoints. The existing Control Catalog endpoints supporting IPv4 remains available for backwards compatibility. The new dual-stack domains are available either from the internet or from within an Amazon Virtual Private Cloud (VPC) using AWS PrivateLink.

January - December 2024

In 2024, AWS Control Tower released the following updates:

- AWS Control Tower CfCT supports GitHub and RCPs
- AWS Control Tower adds preventive controls with declarative policies
- AWS Control Tower adds prescriptive backup plan options
- AWS Control Tower integrates AWS Config controls
- AWS Control Tower improves hook management and adds proactive control Regions
- AWS Control Tower launches managed resource control policies
- AWS Control Tower reports control policy drift
- New `ResetEnabledControl` API
- Control catalog updates `GetControl` API
- AWS Control Tower AFT supports GitLab
- AWS Control Tower available in AWS Asia Pacific (Malaysia) Region
- AWS Control Tower supports up to 1000 accounts per OU
- AWS Control Tower adds landing zone version selection
- Descriptive control API available, expanded access to Regions and controls

- AWS Control Tower supports AFT and CfCT in opt-in Regions
- AWS Control Tower adds the `ListLandingZoneOperations` API
- AWS Control Tower supports up to 100 concurrent control operations
- AWS Control Tower available in AWS Canada West (Calgary)
- AWS Control Tower supports self-service quota adjustments
- AWS Control Tower releases the *Controls Reference Guide*
- AWS Control Tower updates and renames two proactive controls
- Deprecated controls no longer available
- AWS Control Tower supports tagging `EnabledControl` resources in AWS CloudFormation
- AWS Control Tower supports APIs for OU registration and configuration with baselines

AWS Control Tower CfCT supports GitHub and RCPs

December 9, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports GitHubTM™ as an option for a third-party version control system (VCS) and configuration source for Customizations for AWS Control Tower (CFCT). For more infomration, see Set up GitHub as the configuration source.

AWS Control Tower now supports resource control policies (RCPs) for Customizations for AWS Control Tower (CFCT). For more information, see CfCT customization guide.

AWS Control Tower adds preventive controls with declarative policies

December 1, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports preventive controls that are implemented by declarative policies from AWS Organizations. Declarative policies are applied directly at the service level. This approach ensures that the specified configuration is enforced, even when new features or APIs are introduced by the service. For more information, see Controls implemented with declarative policies.

AWS Control Tower adds prescriptive backup plan options

November 25, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports prescriptive AWS Backup plans that allow you to incorporate a data backup and recovery workflow directly into your landing zone. The backup plan includes predefined rules, such as retention days, backup frequency, and the time window during which backup occurs. These rules define how to back up your AWS resources across all of your governed member accounts. When you apply a backup plan to the landing zone, AWS Control Tower ensures that the plan is consistent for all member accounts, and aligned with best practice recommendations from AWS Backup.

For more information, see AWS Backup and AWS Control Tower.

AWS Control Tower integrates AWS Config controls

November 21, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has integrated selected AWS Config controls, so that they can be viewed and managed by AWS Control Tower.

For more information, see Integrated AWS Config controls available in AWS Control Tower

AWS Control Tower improves hook management and adds proactive control Regions

November 20, 2024

(No update required for AWS Control Tower landing zone.)

With this release, you can utilize the full capacity of AWS CloudFormation hooks, without restriction by AWS Control Tower. Also, proactive controls are available in the Canada West (Calgary) Region and Asia Pacific (Malaysia) Region.

Previously, all AWS CloudFormation hooks in your environment needed to be protected by the **CT.CLOUDFORMATION.PR.1** control, so that only AWS Control Tower could modify them. With this release, you can deploy AWS CloudFormation hooks, and modify these hooks, without the restrictions that previously were required by the AWS Control Tower service.

If you currently deploy proactive controls, you can move to this improved hook functionality. To reset all proactive controls on an OU, reset any single proactive control that is active on that

OU. You can perform the reset by calling the `ResetEnabledControl` API, or by updating the control from the console, with the **Reset** functionality. When you complete this reset task for any proactive control, AWS Control Tower moves all of the proactive control hooks on the OU to the new capability. Repeat this process for each OU that deploys proactive controls.

After you reset any proactive control, remove the **CT.CLOUDFORMATION.PR.1** control on your AWS Control Tower OUs, unless you enabled that control for another purpose. If you don't turn off the **CT.CLOUDFORMATION.PR.1** control, you won't be able to create and modify your other AWS CloudFormation hooks.

For more information, see [Update proactive control hooks](#).

AWS Control Tower launches managed resource control policies

November 15, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower offers a new type of preventive control, implemented with resource control policies (RCPs). These controls help you establish a data perimeter across your AWS Control Tower environment, to protect your resources from unintended access.

For example, you can enable RCP-based controls for Amazon S3, AWS Security Token Service, AWS Key Management Service, Amazon SQS, and AWS Secrets Manager services. An RCP-based control can enforce a requirement such as "Require that the organization's Amazon S3 resources are accessible only by IAM principals that belong to the organization, or by an AWS service," regardless of the permissions granted on individual bucket policies.

You can configure the new RCP-based controls, and certain existing SCP-based preventive controls, to specify AWS IAM exemptions for principals and resources. If you don't want a principal or a resource to be governed by the control, you can configure an exemption.

By combining preventive, proactive, and detective controls in AWS Control Tower, you can monitor whether your multi-account AWS environment is secure and managed in accordance with best practices, such as the [AWS Foundational Security Best Practices standard](#).

These new RCP-based preventive controls are available in AWS Regions where AWS Control Tower is available. For a full list of AWS Regions where AWS Control Tower is available, see the [AWS Region Table](#).

AWS Control Tower reports control policy drift

November 15, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now reports control policy drift, for controls implemented with resource control policies (RCPs), and controls that are part of the **Security Hub Service-managed Standard: AWS Control Tower**. This type of drift can be remediated through the new `ResetEnabledControl` API. For more information, see [Types of governance drift](#).

New `ResetEnabledControl` API

November 14, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower announces a new API to help you manage control drift programmatically. You can repair control drift and reset a control to its intended configuration. The `ResetEnabledControl` API works with optional AWS Control Tower controls, including **Strongly recommended** and **Elective** controls.

Control exceptions

- Controls that are implemented with service control policies (SCPs) cannot be reset with this API. For more information, see [`ResetEnabledControl`](#).
- Mandatory controls cannot be reset, because they protect AWS Control Tower resources.
- The **Region deny** control for the landing zone must be reset through the console.

Control drift occurs when an AWS Control Tower control is modified outside AWS Control Tower, for example from the AWS Organizations console. Resolving drift helps to ensure your compliance with governance requirements.

Control catalog updates `GetControl` API

November 8, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports an updated `GetControl` API that includes two new fields: `Implementation` types for all controls, and `Parameters` for certain controls that can be configured.

The `GetControl` API is part of the `controlcatalog` namespace of AWS Control Tower.

For more information, see the `GetControl` API in the *Control Catalog API Reference*.

This release includes related changes that are shown in the AWS Control Tower console.

- All existing AWS Security Hub controls have their `Implementation` parameter value changed from AWS Config rule to AWS Security Hub. The corresponding console help panel is modified to reflect this change.
- All existing Hook controls have their `Implementation` parameter value changed from AWS CloudFormation guard rule to AWS CloudFormation hook. The corresponding console help panel is modified to reflect this change.

AWS Control Tower AFT supports GitLab

October 23, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports GitLabTM™ and GitLab Self-managed as options for a third-party version control system (VCS) and configuration source for Account Factory for Terraform (AFT).

AWS Control Tower available in AWS Asia Pacific (Malaysia) Region

October 21, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in the AWS Asia Pacific (Malaysia) Region.

For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower supports up to 1000 accounts per OU

August 30, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has increased the maximum number of accounts allowed per organizational unit (OU) from 300 to 1000. Now, you can enroll up to 1000 AWS accounts into AWS Control Tower governance at once, without changing your OU structure. The OU registration and re-registration processes also are more efficient, requiring significantly less time to deploy AWS Control Tower baseline resources into your accounts.

Some account limitations still apply due to the limitations on the number of AWS CloudFormation stack sets available. Specifically, the maximum number of accounts you can enroll in an OU may differ, depending on the number of Regions you have under governance. To learn more, visit Limitations based on underlying AWS services in the *AWS Control Tower User Guide*. For a full list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds landing zone version selection

August 15, 2024

(No update required for AWS Control Tower landing zone.)

If you are running AWS Control Tower landing zone version 3.1 and above, you can update or repair your landing zone in place on the current version, or you can upgrade to a version of your choice. Previously, any landing zone update or repair required an upgrade to the latest landing zone version.

With landing zone version selection, you have more flexibility to plan for version upgrades while you evaluate potential changes to your environment. You need not choose between repairing drift to stay in compliance, updating your landing zone configurations, or upgrading to the latest landing zone version. If you are running landing zone version 3.1 or above, you can choose to stay on the current version, or upgrade to a newer version, when you update or reset your landing zone configurations.

Descriptive control API available, expanded access to Regions and controls

August 6, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower added two new API operations that help you find more information about available controls, programmatically. This functionality makes it easier to deploy controls with automation.

- The `GetControl` API returns details about an enabled control, including the target identifier, a control information summary, a list of target Regions, and the drift status.
- The `ListControls` API returns a paginated list of all available controls in the AWS Control Tower library of controls.

These APIs are reached through the AWS Control Catalog namespace. The AWS Control Catalog is a part of AWS Control Tower, which includes controls that help you manage other AWS services, not just AWS Control Tower. This expanded catalog consolidates controls from several AWS services, so that you can view AWS controls according to some common use cases, such as: security, cost, durability, and operations. For more information, see the Control Catalog API Reference.

Expanded Region availability

Beginning with this release, you can extend AWS Control Tower governance into AWS Regions where some of your (already) enabled controls are not available. Also, you can now enable certain controls in more Regions, even though the control is not supported in all of your governed Regions.

Previously, AWS Control Tower prevented you from extending governance into Regions or enabling controls, when it did not offer consistency across all of your enabled controls and governed Regions. With this release, you have more flexibility, as well as more responsibility to ensure that your configuration is correct for all enabled controls and all governed Regions. The AWS Control Tower control APIs and the control catalog APIs can help you get information about the AWS Regions in which you are protected by enabled controls, and the Regions in which additional controls may be deployed. Region and control information also is available in the AWS Control Tower console.

AWS Control Tower supports AFT and CfCT in opt-in Regions

July 18, 2024

(No update required for AWS Control Tower landing zone.)

Today, AWS Control Tower customization frameworks Account Factory for Terraform (AFT) and Customizations for AWS Control Tower (CfCT) are available in five additional AWS Regions: Asia Pacific (Hyderabad, Jakarta and Osaka), Israel (Tel Aviv), and Middle East (UAE).

Account Factory for Terraform (AFT) sets up a Terraform pipeline to help you provision and customize accounts in AWS Control Tower. Customizations for AWS Control Tower (CfCT) helps you customize your AWS Control Tower landing zone and accounts with AWS CloudFormation templates and service control policies (SCPs).

To learn more, visit the Account Factory for Terraform and Customizations for AWS Control Tower pages; in the AWS Control Tower User Guide. You also may wish to review the release notes on the AFT Github page and the CfCT Github page. AFT and CfCT are supported in all AWS Regions, with some exceptions. For specifics, see Region limitations.

AWS Control Tower adds the `ListLandingZoneOperations` API

June 26, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added an API that allows you to retrieve a list of operations recently applied to your landing zone, and operations currently in progress. The API can return the history of landing zone operations and their identifiers for up to 90 days. For usage examples, see View the status of your landing zone operations.

For more information about the `ListLandingZoneOperations` API, see `ListLandingZoneOperations` in the *AWS Control Tower API Reference*.

AWS Control Tower supports up to 100 concurrent control operations

May 20, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports multiple control operations with higher concurrency. You can submit up to 100 AWS Control Tower control operations, across multiple organizational units (OUs), at the same time, from the console or with APIs. Up to ten (10) operations can run simultaneously, and the additional ones are queued. In this way, you can set up a more standardized configuration across multiple AWS accounts, without the operational burden of repetitive control operations.

To monitor the status of your ongoing and queued control operations, you can navigate to the new **Recent operations** page in the AWS Control Tower console, or you can call the new `ListControlOperations` API.

The AWS Control Tower library contains more than 500 controls, which map to different control objectives, frameworks, and services. For a specific control objective, such as **Encrypt data at rest**, you can enable multiple controls with a single control operation, to help you achieve the objective. This capability facilitates accelerated development, allows faster adoption of best practice controls, and mitigates operational complexities.

AWS Control Tower available in AWS Canada West (Calgary)

May 3, 2024

(No update required for AWS Control Tower landing zone.)

Starting today, you can activate AWS Control Tower in the Canada West (Calgary) Region. If you already have deployed AWS Control Tower and you want to extend its governance features to this Region, you can do so with the AWS Control Tower landing zone APIs. Or from the console, go to the **Settings** page in your AWS Control Tower dashboard, select your Regions, and then update your landing zone.

The Canada West (Calgary) Region does not support AWS Service Catalog. For this reason, some functionality of AWS Control Tower is different. The most notable functionality change is that Account Factory is not available. If you choose Canada West (Calgary) as your home Region, the procedures for updating accounts, setting up account automations, and any other processes that involve Service Catalog are different than in other Regions.

Provisioning accounts

To create and provision a new account in the Canada West (Calgary) Region, we recommend that you create an account outside of AWS Control Tower, and then enroll it into a registered OU. For more information, see Enroll an existing account and Steps to enroll an account.

The Service Catalog APIs are not available in Canada West (Calgary) Region. The example script shown in Automate account provisioning in AWS Control Tower by Service Catalog APIs is not workable.

Account Factory Customizations (AFC), Account Factory for Terraform (AFT), and Customizations for AWS Control Tower (CfCT) are not available in Canada West (Calgary), due to lack of other underlying dependencies for AWS Control Tower. If you extend governance to Canada West (Calgary) Region, you can continue to manage AFC blueprints in all Regions that AWS Control Tower supports, as long as Service Catalog is available in your home Region.

Controls

Proactive controls and controls for the **AWS Security Hub Service-Managed Standard: AWS Control Tower** are not available in Canada West (Calgary) Region. The preventive control `CT.CLOUDFORMATION.PR.1` is not available in Canada West (Calgary) because it is required only for activating the hook-based, proactive controls. Certain detective controls based on AWS Config are not available. For details, see Control limitations.

Identity provider

IAM Identity Center is not available in Canada West (Calgary). The best practice recommendation is to set up your landing zone in a Region where IAM Identity Center is available. Alternatively, you have the option to self-manage your account access configuration if you use an external identity provider in Canada West (Calgary).

The unavailability of Service Catalog in Canada West (Calgary) Region has no effect on other Regions that are supported by AWS Control Tower. These differences apply only if your home Region is Canada West (Calgary).

For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower supports self-service quota adjustments

April 25, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports self-service quota adjustments through the Service Quotas console. For more information, see Request a quota increase.

AWS Control Tower releases the *Controls Reference Guide*

April 21, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower released the *Controls Reference Guide*, a new document where you can find detailed information about the controls that are specific to the AWS Control Tower environment. Previously, this material was included in the *AWS Control Tower User Guide*. The *Controls Reference*

Guide covers controls in an expanded format. For more information, see the AWS Control Tower Controls Reference Guide.

AWS Control Tower updates and renames two proactive controls

March 26, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has renamed two proactive controls to align with updates to Amazon OpenSearch Service.

- **[CT.OPENSEARCH.PR.8]** Require an Elasticsearch Service domain to use TLSv1.2
- **[CT.OPENSEARCH.PR.16]** Require an Amazon OpenSearch Service domain to use TLSv1.2

We updated the control names and the artifacts for these two controls to align with the recent release from the Amazon OpenSearch Service, which now supports Transport Layer Security (TLS) version 1.3 among its transport security options for domain endpoint security.

To add support for TLSv1.3 for these controls, we have updated the artifact and name of the controls to reflect the intent of the control. They now evaluate the minimum TLS version of the service domain. To make this update in your environment, you must **Disable** and **Enable** the controls to deploy the latest artifact.

No other proactive controls are affected by this change. We recommend that you review these controls, to ensure that they meet your control objectives.

For questions or concerns, contact AWS Support.

Deprecated controls no longer available

March 12, 2024

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has deprecated some controls. These controls are no longer available.

- CT.ATHENA.PR.1
- CT.CODEBUILD.PR.4

- CT.AUTOSCALING.PR.3
- SH.Athena.1
- SH.Codebuild.5
- SH.AutoScaling.4
- SH.SNS.1
- SH.SNS.2

AWS Control Tower supports tagging `EnabledControl` resources in AWS CloudFormation

February 22, 2024

(No update required for AWS Control Tower landing zone.)

This AWS Control Tower release updates the behavior of the `EnabledControl` resource, to align better with configurable controls, and to improve the ability to manage your AWS Control Tower environment with automation. With this release, you can add tags to configurable `EnabledControl` resources by means of AWS CloudFormation templates. Previously, you could add tags through the AWS Control Tower console and APIs only.

The AWS Control Tower `GetEnabledControl`, `EnableControl`, and `ListTagsforResource` API operations are updated with this release, because they rely on the `EnabledControl` resource functionality.

For more information, see [Tagging `EnabledControl` resources in AWS Control Tower] and [`EnabledControl`] in the *AWS CloudFormation User Guide*.

AWS Control Tower supports APIs for OU registration and configuration with baselines

February 14, 2024

(No update required for AWS Control Tower landing zone.)

These APIs support programmatic OU registration with the `EnableBaseline` call. When you enable a baseline on an OU, member accounts within the OU are enrolled into AWS Control Tower governance. Certain caveats may apply. For example, OU registration through the AWS Control Tower console enables optional controls as well as mandatory controls. When calling APIs, you may need to complete an extra step so that optional controls are enabled.

An AWS Control Tower *baseline* embodies best practices for AWS Control Tower governance of an OU and member accounts. For example, when you enable a baseline on an OU, member accounts within the OU receive a defined group of resources, including AWS CloudTrail, AWS Config, IAM Identity Center, and required AWS IAM roles.

Specific baselines are compatible with specific AWS Control Tower landing zone versions. AWS Control Tower can apply the latest compatible baseline to your landing zone, when you change your landing zone settings. For more information, see Compatibility of OU baselines and landing zone versions.

This release includes four essential Types of baselines

- `AWSControlTowerBaseline`
- `AuditBaseline`
- `LogArchiveBaseline`
- `IdentityCenterBaseline`

With the new APIs and defined baselines, you can register OUs and automate your OU provisioning workflow. The APIs also can manage OUs that are already under AWS Control Tower governance, so you can re-register OUs after landing zone updates. The APIs include support for an AWS CloudFormation `EnabledBaseline` resource, which allows you to manage your OUs with infrastructure as code (IaC).

Baseline APIs

- **EnableBaseline**, **UpdateEnabledBaseline**, **DisableBaseline**: Take action on a baseline for an OU.
- **GetEnabledBaseline**, **ListEnabledBaselines**: Discover configurations for your enabled baselines.
- **GetBaselineOperation**: View the status of a particular baseline operation.
- **ResetEnabledBaseline**: Remediate resource drift on an OU with an enabled baseline (including nested OUs and mandatory control drift). Also remediates drift for the landing-zone-level Region deny control
- **GetBaseline**, **ListBaselines**: Discover content of AWS Control Tower baselines.

To learn more about these APIs, review Baselines in the AWS Control Tower User Guide, and the API Reference. The new APIs are available in AWS Regions where AWS Control Tower is available, except GovCloud (US) Regions. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

January - December 2023

In 2023, AWS Control Tower released the following updates:

- AWS Control Tower adds new proactive controls
- AWS Control Tower updates Amazon EC2 controls
- AWS Control Tower available in seven additional AWS Regions
- AWS Control Tower Account Factory Customization (AFC) and request tracing generally available
- AWS Control Tower landing zone version 3.1
- AWS Control Tower proactive controls generally available

Transition to new AWS Service Catalog External product type (phase 3)

December 14, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower no longer supports *Terraform Open Source* as a product type (blueprint) when creating new AWS accounts. For more information and for instructions about updating your account blueprints, review Transition to the AWS Service Catalog External product type.

If you do not update your account blueprints to use the *External* product type, you can only update or terminate accounts that you provisioned using Terraform Open Source blueprints.

AWS Control Tower landing zone version 3.3

December 14, 2023

(Update required for AWS Control Tower landing zone to version 3.3. For information, see Update your landing zone).

Updates to S3 bucket policy in the AWS Control Tower Audit account

We have modified the Amazon S3 Audit bucket policy that AWS Control Tower deploys in accounts, so that an `aws:SourceOrgID` condition must be met for any write permissions. With this release, AWS services have access to your resources only when the request originates from your organization or organizational unit (OU).

You can use the `aws:SourceOrgID` condition key and set the value to your **organization ID** in the condition element of your S3 bucket policy. This condition ensures that CloudTrail only can write logs on behalf of accounts within your organization to your S3 bucket; it prevents CloudTrail logs outside your organization from writing to your AWS Control Tower S3 bucket.

We made this change to remediate a potential security vulnerability, without affecting the functionality of your existing workloads. To view the updated policy, see Amazon S3 bucket policy in the audit account.

For more information about the new condition key, see the IAM documentation and the IAM blog post entitled "*Use scalable controls for AWS services accessing your resources*."

Updates to the policy in the AWS Config SNS topic

We added the new `aws:SourceOrgID` condition key to the policy for the AWS Config SNS topic.To view the updated policy, see The AWS Config SNS topic policy.

Updates to the landing zone Region Deny control

- Removed `discovery-marketplace:`. This action is covered by the `aws-marketplace:*` exemption.
- Added `quicksight:DescribeAccountSubscription`

Updated AWS CloudFormation template

We updated the AWS CloudFormation template for the stack named `BASELINE-CLOUDTRAIL-MASTER` so that is does not show drift when AWS KMS encryption is not used.

Transition to new AWS Service Catalog External product type (phase 2)

December 7, 2023

(No update required for AWS Control Tower landing zone.)

HashiCorp updated their Terraform licensing. As a result, AWS Service Catalog changed support for *Terraform Open Source* products and provisioned products to a new product type, called *External*.

To avoid disruption to existing workloads and AWS resources in your accounts, follow the AWS Control Tower transition steps in Transition to the AWS Service Catalog External product type by December 14, 2023.

AWS Control Tower announces controls to assist digital sovereignty

November 27, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower announces 65 new AWS-managed controls, to help you meet your digital sovereignty requirements. With this release, you can discover these controls under a new *digital sovereignty group* in the AWS Control Tower console. You can use these controls to help prevent actions and detect resource changes regarding *data residency*, *granular access restriction*, *encryption*, and *resiliency* capabilities. These controls are designed to make it simpler for you to address requirements at scale. For more information about digital sovereignty controls, see Controls that enhance digital sovereignty protection.

For example, you can choose to enable controls that help enforce your encryption and resiliency strategies, such as **Require an AWS AppSync API cache to have encryption in transit enabled** or **Require an AWS Network Firewall to be deployed across multiple Availability Zones**. You can also customize the AWS Control Tower Region deny control to apply regional restrictions that best fit your unique business needs.

This release brings well-enhanced AWS Control Tower Region deny capabilities. You can apply a new, parameterized Region deny control at the OU level, for increased granularity of governance, while maintaining additional Region governance at the landing zone level. This customizable Region deny control helps you to apply regional restrictions that best fit your unique business needs. For more information about the new, configurable Region deny control, see Region deny control applied to the OU.

As a new tool to the new Region deny enhancement, this release includes a new API, `UpdateEnabledControl`, which allows you to reset your enabled controls to the default settings. This API is especially helpful in use cases where you need to resolve drift quickly, or to guarantee programmatically that a control is not in a state of drift. For more information about the new API, see the AWS Control Tower API Reference

New proactive controls

- CT.APIGATEWAY.PR.6: Require an Amazon API Gateway REST domain to use a security policy that specifies a minimum TLS protocol version of TLSv1.2
- CT.APPSYNC.PR.2: Require an AWS AppSync GraphQL API to be configured with private visibility
- CT.APPSYNC.PR.3: Require that an AWS AppSync GraphQL API is not authenticated with API keys
- CT.APPSYNC.PR.4: Require an AWS AppSync GraphQL API cache to have encryption in transit enabled.
- CT.APPSYNC.PR.5: Require an AWS AppSync GraphQL API cache to have encryption at rest enabled.

- CT.AUTOSCALING.PR.9: Require an Amazon EBS volume configured through an Amazon EC2 Auto Scaling launch configuration to encrypt data at rest
- CT.AUTOSCALING.PR.10: Require an Amazon EC2 Auto Scaling group to use only AWS Nitro instance types when overriding a launch template
- CT.AUTOSCALING.PR.11: Require only AWS Nitro instance types that support network traffic encryption between instances to be added to an Amazon EC2 Auto Scaling group, when overriding a launch template
- CT.DAX.PR.3: Require an DynamoDB Accelerator cluster to encrypt data in transit with Transport Layer Security (TLS)
- CT.DMS.PR.2: Require an AWS Database Migration Service (DMS) Endpoint to encrypt connections for source and target endpoints
- CT.EC2.PR.15: Require an Amazon EC2 instance to use an AWS Nitro instance type when creating from the `AWS::EC2::LaunchTemplate` resource type
- CT.EC2.PR.16: Require an Amazon EC2 instance to use an AWS Nitro instance type when created using the `AWS::EC2::Instance` resource type
- CT.EC2.PR.17: Require an Amazon EC2 dedicated host to use an AWS Nitro instance type
- CT.EC2.PR.18: Require an Amazon EC2 fleet to override only those launch templates with AWS Nitro instance types
- CT.EC2.PR.19: Require an Amazon EC2 instance to use a nitro instance type that supports encryption in-transit between instances when created using the `AWS::EC2::Instance` resource type
- CT.EC2.PR.20: Require an Amazon EC2 fleet to override only those launch templates with AWS Nitro instance types that support encryption in transit between instances
- CT.ELASTICACHE.PR.8: Require an Amazon ElastiCache replication group of later Redis versions to have RBAC authentication activated
- CT.MQ.PR.1: Require an Amazon MQ ActiveMQ broker to use use active/standby deployment mode for high availability
- CT.MQ.PR.2: Require an Amazon MQ Rabbit MQ broker to use Multi-AZ cluster mode for high availability
- CT.MSK.PR.1: Require an Amazon Managed Streaming for Apache Kafka (MSK) cluster to enforce encryption in transit between cluster broker nodes
- CT.MSK.PR.2: Require an Amazon Managed Streaming for Apache Kafka (MSK) cluster to be configured with PublicAccess disabled

- CT.NETWORK-FIREWALL.PR.5: Require an AWS Network Firewall firewall to be deployed across multiple Availability Zones
- CT.RDS.PR.26: Require an Amazon RDS DB Proxy to require Transport Layer Security (TLS) connections
- CT.RDS.PR.27: Require an Amazon RDS DB cluster parameter group to require Transport Layer Security (TLS) connections for supported engine types
- CT.RDS.PR.28: Require an Amazon RDS DB parameter group to require Transport Layer Security (TLS) connections for supported engine types
- CT.RDS.PR.29: Require an Amazon RDS cluster not be configured to be publicly accessible by means of the 'PubliclyAccessible' property
- CT.RDS.PR.30: Require that an Amazon RDS database instance has encryption at rest configured to use a KMS key that you specify for supported engine types
- CT.S3.PR.12: Require an Amazon S3 access point to have a Block Public Access (BPA) configuration with all options set to true

New preventive controls

- CT.APPSYNC.PV.1 Require that an AWS AppSync GraphQL API is configured with private visibility
- CT.EC2.PV.1 Require an Amazon EBS snapshot to be created from an encrypted EC2 volume
- CT.EC2.PV.2 Require that an attached Amazon EBS volume is configured to encrypt data at rest
- CT.EC2.PV.3 Require that an Amazon EBS snapshot cannot be publicly restorable
- CT.EC2.PV.4 Require that Amazon EBS direct APIs are not called
- CT.EC2.PV.5 Disallow the use of Amazon EC2 VM import and export
- CT.EC2.PV.6 Disallow the use of deprecated Amazon EC2 RequestSpotFleet and RequestSpotInstances API actions
- CT.KMS.PV.1 Require an AWS KMS key policy to have a statement that limits creation of AWS KMS grants to AWS services
- CT.KMS.PV.2 Require that an AWS KMS asymmetric key with RSA key material used for encryption does not have a key length of 2048 bits
- CT.KMS.PV.3 Require that an AWS KMS key is configured with the bypass policy lockout safety check enabled
- CT.KMS.PV.4 Require that an AWS KMS customer-managed key (CMK) is configured with key material originating from AWS CloudHSM

- CT.KMS.PV.5 Require that an AWS KMS customer-managed key (CMK) is configured with imported key material
- CT.KMS.PV.6 Require that an AWS KMS customer-managed key (CMK) is configured with key material originating from an external key store (XKS)
- CT.LAMBDA.PV.1 Require an AWS Lambda function URL to use AWS IAM-based authentication
- CT.LAMBDA.PV.2 Require an AWS Lambda function URL to be configured for access only by principals within your AWS account

AWS Control Tower landing zone APIs

November 26, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now offers APIs that help you manage your landing zone programmatically. These APIs allow you to create, update, and reset your landing zone, as well as retrieve information about your landing zone configuration and operations. For more information, see Landing zone API examples.

The landing zone APIs are available in all AWS Regions where AWS Control Tower is available, except GovCloud (US) Regions. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower control tagging APIs

November 10, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now offers APIs that help you tag your enabled controls programmatically. These APIs allow you to add, remove, and list tags for your enabled controls. For more information, see Tagging AWS Control Tower resources.

The control tagging APIs are available in all AWS Regions where AWS Control Tower is available, except GovCloud (US) Regions. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in AWS Asia Pacific (Melbourne)

November 3, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in the Asia Pacific (Melbourne). For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

Transition to new AWS Service Catalog External product type (phase 1)

October 31, 2023

(No update required for AWS Control Tower landing zone.)

HashiCorp updated their Terraform licensing. As a result, AWS Service Catalog changed support for *Terraform Open Source* products and provisioned products to a new product type, called *External*.

To avoid disruption to existing workloads and AWS resources in your accounts, follow the AWS Control Tower transition steps in Transition to the AWS Service Catalog External product type by December 14, 2023.

AWS Control Tower adds new control API

October 27, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now offers a new API, `UpdateEnabledControl`, which allows you to update your enabled controls. This API is especially helpful in use cases where you need to resolve drift quickly, or to guarantee programmatically that a control is not in a state of drift. For more information about the new API, see the AWS Control Tower API Reference.

The `UpdateEnabledControl` API is available in all AWS Regions where AWS Control Tower is available, except GovCloud (US) Regions. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds new controls

October 20, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added 22 new controls to the AWS Control Tower library of controls. These controls help you enforce best practices for your AWS resources. For more information about the new controls, see Control categories.

The new controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower detects trusted access drift

October 13, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now detects and reports drift for trusted access settings. Trusted access settings allow AWS Control Tower to interact with other AWS services on your behalf. If these settings are changed outside of AWS Control Tower, AWS Control Tower will detect the drift and report it in the AWS Control Tower console. For more information about trusted access drift, see Types of governance drift.

Trusted access drift detection is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in four additional AWS Regions

September 29, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in four additional AWS Regions: Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Europe (Spain), and Europe (Zurich). For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in AWS Israel (Tel Aviv)

August 1, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in the Israel (Tel Aviv). For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds 28 new proactive controls

July 27, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added 28 new proactive controls to the AWS Control Tower library of controls. These controls help you enforce best practices for your AWS resources. For more information about the new controls, see Control categories.

The new controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower deprecates two controls

July 27, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has deprecated two controls: CT.CLOUDFORMATION.PR.2 and CT.CLOUDFORMATION.PR.3. These controls are no longer available in the AWS Control Tower library of controls. For more information about the deprecated controls, see Control categories.

The deprecated controls are no longer available in any AWS Region.

AWS Control Tower landing zone version 3.2

June 16, 2023

(Update required for AWS Control Tower landing zone to version 3.2. For information, see Update your landing zone).

AWS Control Tower landing zone version 3.2 brings the controls that are part of the AWS Security Hub **Service-Managed Standard: AWS Control Tower** to general availability. It introduces the ability to view the drift status of controls that are part of this standard in the AWS Control Tower console.

This update includes a new service-linked role (SLR), called the **AWSServiceRoleForAWSControlTower**. This role assists AWS Control Tower by creating an EventBridge Managed Rule, called the **AWSControlTowerManagedRule** in each member account. This managed rule collects AWS Security Hub **Finding** events, from with AWS Control Tower can determine control drift.

This rule is the first managed rule to be created by AWS Control Tower. The rule is not deployed by a stack; it is deployed directly from the EventBridge APIs. You can view the rule in the EventBridge console, or by means of the EventBridge APIs. If the `managed-by` field is populated, it will show the AWS Control Tower service principal.

Previously, AWS Control Tower assumed the **AWSControlTowerExecution** role to perform operations in member accounts. This new role and rule are better aligned with the best practices principle of allowing least privilege when performing operations in a multi-account AWS environment. The new role provides scoped-down permissions that specifically allow: creating the managed rule in member accounts, maintaining the managed rule, publishing security notifications through SNS, and verifying drift. For more information, see AWSServiceRoleForAWSControlTower.

The landing zone 3.2 update also includes a new StackSet resource in the management account, `BP_BASELINE_SERVICE_LINKED_ROLE`, which initially deploys the service-linked role.

When reporting Security Hub control drift (in landing zone 3.2 and later), AWS Control Tower receives a daily status update from Security Hub. Although controls are active in every governed Region, AWS Control Tower sends the AWS Security Hub **Finding** events to the AWS Control Tower home Region only. For more information, see Security Hub control drift reporting.

Update to the Region Deny control

This landing zone version also includes an update to the Region Deny control.

Global services and APIs added

- AWS Billing and Cost Management (`billing:*`)
- AWS CloudTrail (`cloudtrail:LookupEvents`) to allow visibility of global events in member accounts.
- AWS Consolidated Billing (`consolidatedbilling:*`)
- AWS Management Console Mobile Application (`consoleapp:*`)
- AWS Free Tier (`freetier:*`)
- AWS Invoicing (`invoicing:*`)
- AWS IQ (`iq:*`)
- AWS User Notifications (`notifications:*`)
- AWS User Notifications Contacts (`notifications-contacts:*`)
- Amazon Payments (`payments:*`)
- AWS Tax Settings (`tax:*`)

Global services and APIs removed

- Removed `s3:GetAccountPublic` because it is not a valid action.
- Removed `s3:PutAccountPublic` because it is not a valid action.

AWS Control Tower adds email-to-ID mapping for IAM Identity Center

July 13, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports email-to-ID mapping for IAM Identity Center. This feature allows you to map email addresses to IAM Identity Center user IDs, which makes it easier to manage user access to your AWS Control Tower environment. For more information about email-to-ID mapping, see Integration with IAM Identity Center.

Email-to-ID mapping is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds more AWS Security Hub controls

June 29, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added more AWS Security Hub controls to the AWS Control Tower library of controls. These controls help you enforce best practices for your AWS resources. For more information about the new controls, see Control categories.

The new controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower publishes metadata for AWS Security Hub controls

June 22, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now publishes metadata for AWS Security Hub controls. This metadata includes information about the control, such as the control ID, control title, and control description. For more information about the metadata, see Control metadata.

Control metadata is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds Account Factory Customization (AFC) for Terraform

June 15, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports Account Factory Customization (AFC) for Terraform. This feature allows you to use Terraform to customize your AWS Control Tower accounts. For more information about AFC for Terraform, see Account Factory Customization for Terraform.

AFC for Terraform is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds self-managed IAM identity center

June 8, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports self-managed IAM identity center. This feature allows you to use your own identity provider with AWS Control Tower. For more information about self-managed IAM identity center, see IAM identity center.

Self-managed IAM identity center is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds mixed governance note

June 1, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now includes a note about mixed governance. This note explains how AWS Control Tower works with other AWS services to provide governance for your AWS resources. For more information about mixed governance, see Mixed governance.

The mixed governance note is available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower adds new proactive controls

May 25, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added new proactive controls to the AWS Control Tower library of controls. These controls help you enforce best practices for your AWS resources. For more information about the new controls, see Control categories.

The new controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower updates Amazon EC2 controls

May 18, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has updated the Amazon EC2 controls in the AWS Control Tower library of controls. These updates improve the security and reliability of your AWS Control Tower environment. For more information about the updated controls, see Control categories.

The updated controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in seven additional AWS Regions

May 11, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in seven additional AWS Regions: Asia Pacific (Osaka), Canada (Central), Europe (Milan), Europe (Stockholm), Middle East (Bahrain), Middle East (UAE), and South America (São Paulo). For a full list of Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower Account Factory Customization (AFC) and request tracing generally available

April 27, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower Account Factory Customization (AFC) and request tracing are now generally available. AFC allows you to customize your AWS Control Tower accounts, and request tracing allows you to track the status of your AWS Control Tower requests. For more information about AFC and request tracing, see Account Factory Customization and Request tracing.

AFC and request tracing are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower landing zone version 3.1

February 9, 2023

(Update required for AWS Control Tower landing zone to version 3.1. For information, see Update your landing zone)

AWS Control Tower landing zone version 3.1 includes the following updates:

- With this release, AWS Control Tower deactivates unnecessary access logging for your *access logging bucket*, which is the Amazon S3 bucket where access logs are stored in the Log Archive account, while continuing to enable server access logging for S3 buckets. This release also includes updates to the Region Deny control that allow additional actions for global services, such as Support Plans and AWS Artifact.
- Deactivation of server access logging for the AWS Control Tower access logging bucket causes Security Hub to create a finding for the Log Archive account's *access logging bucket*, due to an AWS Security Hub rule, [S3.9] S3 bucket server access logging should be enabled. In alignment with Security Hub, we recommend that you suppress this particular finding, as stated in the Security Hub description of this rule. For additional information, see information about suppressed findings.
- Access logging for the (regular) logging bucket in the Log Archive account is unchanged in version 3.1. In alignment with best practices, access events for that bucket are recorded as log entries in the *access logging bucket*. For more information about access logging, see Logging requests using server access logging in the Amazon S3 documentation.

- We made an update of the Region Deny control. This update allows actions by more global services. For details of this SCP, see Deny access to AWS based on the requested AWS Region and Controls that enhance data residency protection.

  Global services added:

  - AWS Account Management (`account:*`)
  - AWS Activate (`activate:*`)
  - AWS Artifact (`artifact:*`)
  - AWS Billing Conductor (`billingconductor:*`)
  - AWS Compute Optimizer (`compute-optimizer:*`)
  - AWS Data Pipeline (`datapipeline:GetAccountLimits`)
  - AWS Device Farm(`devicefarm:*`)
  - AWS Marketplace (`discovery-marketplace:*`)
  - Amazon ECR (`ecr-public:*`)
  - AWS License Manager (`license-manager:ListReceivedLicenses`)
  - AWS Lightsail (`lightsail:Get*`)
  - AWS Resource Explorer (`resource-explorer-2:*`)
  - Amazon S3 (`s3:CreateMultiRegionAccessPoint`, `s3:GetBucketPolicyStatus`, `s3:PutMultiRegionAccessPointPolicy`)
  - AWS Savings Plans (`savingsplans:*`)
  - IAM Identity Center (`sso:*`)
  - AWS Support App (`supportapp:*`)
  - Support Plans (`supportplans:*`)
  - AWS Sustainability (`sustainability:*`)
  - AWS Resource Groups Tagging API (`tag:GetResources`)
  - AWS Marketplace Vendor Insights (`vendor-insights:ListEntitledSecurityProfiles`)

AWS Control Tower proactive controls generally available

April 13, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower proactive controls are now generally available. Proactive controls help you enforce best practices for your AWS resources. For more information about proactive controls, see Proactive controls.

Proactive controls are available in all AWS Regions where AWS Control Tower is available. For a list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

January - December 2022

In 2022, AWS Control Tower released the following updates:

- Concurrent account operations
- Account Factory Customization (AFC)
- Comprehensive controls assist in AWS resource provisioning and management
- Compliance status viewable for all AWS Config rules
- API for controls and a new AWS CloudFormation resource
- CfCT supports stack set deletion
- Customized log retention
- Role drift repair available
- AWS Control Tower landing zone version 3.0
- The Organization page combines views of OUs and accounts
- Simplified account creation and enrollment
- AFT supports automated customization for shared AWS Control Tower accounts
- Concurrent operations for all optional controls
- Existing security and logging accounts
- AWS Control Tower landing zone version 2.9
- AWS Control Tower landing zone version 2.8

Concurrent account operations

December 16, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports concurrent actions in account factory. You can create, update, or enroll up to five (5) accounts at a time. Submit up to five actions in succession and view the completion status of each request, while your accounts finish building in the background. For example, you no longer must wait for each process to complete before you update another account, or before you re-register an entire organizational unit (OU).

Account Factory Customization (AFC)

November 28, 2022

(No update required for AWS Control Tower landing zone.)

Account factory customization allows you to customize new and existing accounts from within the AWS Control Tower console. These new customization capabilities give you the flexibility to define account blueprints, which are AWS CloudFormation templates contained in a specialized Service Catalog product. Blueprints provision fully customized resources and configurations. You also may choose use pre-defined blueprints, built and managed by AWS partners, that help you customize accounts for specific use cases.

Previously, AWS Control Tower account factory did not support account customization in the console. With this update of account factory, you can pre-define account requirements and implement them as part of a well-defined workflow. You can apply blueprints to create new accounts, to enroll other AWS accounts into AWS Control Tower, and to update existing AWS Control Tower accounts.

When you provision, enroll, or update an account in account factory, you will select the blueprint to deploy. Those resources specified in the blueprint are provisioned in your account. When your account has finished building, all of the custom configurations are available for use immediately.

To get started with customizing accounts, you can define the resources for your intended use case in a Service Catalog product. You also can select partner-managed solutions from the AWS Getting Started Library. For more information, see [Customize accounts with Account Factory Customization (AFC)](#).

Comprehensive controls assist in AWS resource provisioning and management

November 28, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports comprehensive controls management, including new, optional proactive controls, implemented through AWS CloudFormation hooks. These controls are referred to as proactive because they check your resources – before the resources are deployed – to determine whether the new resources will comply with the controls that are activated in your environment.

Over 130 new proactive controls assist you with meeting specific policy objectives for your AWS Control Tower environment; with meeting requirements of industry-standard compliance frameworks; and with governing AWS Control Tower interactions across more than twenty other AWS services.

The AWS Control Tower controls library classifies these controls according to the associated AWS services and resources. For more details, see Proactive controls.

With this release, AWS Control Tower also is integrated with AWS Security Hub, by means of the new Security Hub **Service-Managed Standard: AWS Control Tower**, which supports the AWS Foundational Security Best Practices (FSBP) standard. You can view over 160 Security Hub controls alongside AWS Control Tower controls in the console, and you can obtain an Security Hub security score for your AWS Control Tower environment. For more information, see Security Hub controls.

Compliance status viewable for all AWS Config rules

November 18, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower. You can view the compliance status of all AWS Config rules that affect your accounts in AWS Control Tower, enrolled or unenrolled, without navigating outside of the AWS Control Tower console. Customers can choose to set up Config rules, called detective controls, in AWS Control Tower, or to set them up directly through the AWS Config service. The rules deployed by AWS Config are shown, along with the rules deployed by AWS Control Tower.

Previously, AWS Config rules deployed through the AWS Config service were not visible in the AWS Control Tower console. Customers had to navigate to the AWS Config service to identify non-compliant AWS Config rules. Now you can identify any non-compliant AWS Config rule within the AWS Control Tower console. To view the compliance status of all your Config rules, navigate to the **Account details** page in the AWS Control Tower console. You will see a list showing the compliance status of controls managed by AWS Control Tower and Config rules deployed outside of AWS Control Tower.

API for controls and a new AWS CloudFormation resource

September 1, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports programmatic management of controls, also known as *guardrails*, through a set of API calls. A new AWS CloudFormation resource supports the API functionality for controls. For more details, see Automate tasks in AWS Control Tower and Create AWS Control Tower resources with AWS CloudFormation.

These APIs allow you to enable, disable, and view the application status of controls in the AWS Control Tower library. The APIs include support for AWS CloudFormation, so you can manage AWS resources as infrastructure-as-code (IaC). AWS Control Tower provides optional preventive and detective controls that express your policy intentions regarding an entire organizational unit (OU), and every AWS account within the OU. These rules remain in effect as you create new accounts or make changes to existing accounts.

APIs included in this release

- **EnableControl**– This API call activates a control. It starts an asynchronous operation that creates AWS resources on the specified organizational unit and the accounts it contains.
- **DisableControl**– This API call turns off a control. It starts an asynchronous operation that deletes AWS resources on the specified organizational unit and the accounts it contains.
- **GetControlOperation**– Returns the status of a particular **EnableControl** or **DisableControl** operation.
- **ListEnabledControls**– Lists the controls enabled by AWS Control Tower on the specified organizational unit and the accounts it contains.

To view a list of control names for optional controls, see Resource identifiers for APIs and controls, in the *AWS Control Tower User Guide*.

CfCT supports stack set deletion

August 26, 2022

(No update required for AWS Control Tower landing zone.)

Customizations for AWS Control Tower (CfCT) now supports stack set deletion, by setting a parameter in the `manifest.yaml` file. For more information, see Delete a stack set.

> **Important**
>
> When you initially set the value of `enable_stack_set_deletion` to `true`, the next time you invoke CfCT, **ALL** resources that begin with the prefix `CustomControlTower-`, which have the associated key tag `Key:AWS_Solutions, Value: CustomControlTowerStackSet`, and which are not declared in the manifest file, are staged for deletion.

Customized log retention

August 15, 2022

(Update required for AWS Control Tower landing zone. For information, see Update your landing zone)

AWS Control Tower now provides the ability to customize the retention policy for Amazon S3 buckets that store your AWS Control Tower CloudTrail logs. You can customize your Amazon S3 log retention policy, in increments of days or years, up to a maximum of 15 years.

If you choose not customize your log retention, the default settings are 1 year for standard account logging, and 10 years for access logging.

This feature is available for existing customers through AWS Control Tower when you update or repair your landing zone, and for new customers through the AWS Control Tower setup

Role drift repair available

August 11, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports repair for role drift. You can restore a required role without a full repair of your landing zone. If this type of drift repair is needed, the console error page provides steps for restoring the role, so that your landing zone is once again available.

AWS Control Tower landing zone version 3.0

July 29, 2022

(Update required for AWS Control Tower landing zone to version 3.0. For information, see Update your landing zone)

AWS Control Tower landing zone version 3.0 includes the following updates:

- The option to choose organization-level AWS CloudTrail trails, or to opt out of CloudTrail trails managed by AWS Control Tower.
- Two new detective controls to determine whether AWS CloudTrail is logging activity in your accounts.
- The option to aggregate AWS Config information about global resources in your home Region only.
- An update to the Region deny control.
- An update to the managed policy, **AWSControlTowerServiceRolePolicy**.
- We no longer create the IAM role `aws-controltower-CloudWatchLogsRole` and the CloudWatch log group `aws-controltower/CloudTrailLogs` in each enrolled account. Previously, we created these in each account for its account trail. With organization trails, we only create one in the management account.

The following sections provide more details about each new capability.

Organization-level CloudTrail trails in AWS Control Tower

With landing zone version 3.0, AWS Control Tower now supports organization-level AWS CloudTrail trails.

When you update your AWS Control Tower landing zone to version 3.0, you have the option to select organization-level AWS CloudTrail trails as your logging preference, or to opt out of CloudTrail trails that are managed by AWS Control Tower. When you update to version 3.0, AWS Control Tower deletes *the existing account-level trails for enrolled accounts* after a 24-hour waiting period. AWS Control Tower does not delete account-level trails for unenrolled accounts. In the unlikely case that your landing zone update does not succeed, but the failure occurs after AWS Control Tower already has created the organization-level trail, you may incur duplicate charges for organization-level and account-level trails, until your update operation is able to complete successfully.

Going forward from landing zone 3.0, AWS Control Tower no longer supports account-level trails that AWS manages. Instead, AWS Control Tower creates an organization-level trail, which is active or inactive, according to your selection.

> **Note**
>
> After you update to version 3.0 or later, you do not have the option to continue with account-level CloudTrail trails managed by AWS Control Tower.

No logging data is lost from your aggregated account logs, because the logs remain in the existing Amazon S3 bucket where they are stored. Only the trails are deleted, not the existing logs. If you select the option to add organization-level trails, AWS Control Tower opens a new path to a new folder within your Amazon S3 bucket and continues sending logging information to that location. If you choose to opt out of trails managed by AWS Control Tower, your existing logs remain in the bucket, unchanged.

Path naming conventions for log storage

- Account trail logs are stored with a path of this form: `/org id/AWSLogs/…`
- Organization trail logs are stored with a path of this form: `/org id/AWSLogs/org id/…`

The path that AWS Control Tower creates for your organization-level CloudTrail trails is different than the default path for a manually-created organization-level trail, which would have the following form:

- `/AWSLogs/org id/…`

For more information about CloudTrail path naming, see Finding your CloudTrail log files.

> **Tip**
>
> If you plan to create and manage your own account-level trails, we recommend that you create the new trails before you complete the update to AWS Control Tower landing zone version 3.0, to start logging right away.

At any time, you may choose to create new account-level or organization-level CloudTrail trails and manage them on your own. The option to choose organization-level CloudTrail trails managed by AWS Control Tower is available during any landing zone update to version 3.0 or later. You can opt *into* and opt *out of* organization-level trails, whenever you update your landing zone.

If your logs are managed by a third-party service, be sure to give the new path name to your service.

> **Note**
>
> For landing zones at version 3.0 or later, account-level AWS CloudTrail trails are not supported by AWS Control Tower. You can create and maintain your own account-level trails at any time, or you can opt into organization-level trails managed by AWS Control Tower.

Record AWS Config resources in the home Region only

In landing zone version 3.0, AWS Control Tower has updated the baseline configuration for AWS Config so that it records global resources in the home Region only. After you update to version 3.0, resource recording for global resources is enabled only in your home Region.

This configuration is considered a best practice. It is recommended by AWS Security Hub and AWS Config, and it creates cost savings by reducing the number of configuration items created when global resources are created, modified, or deleted. Previously, each time a global resources was created, updated, or deleted, whether by a customer or by an AWS service, a configuration item was created for each item in each governed Region.

Two new detective controls for AWS CloudTrail logging

As part of the change to organization-level AWS CloudTrail trails, AWS Control Tower is introducing two new detective controls that check whether CloudTrail is enabled. The first control has **Mandatory** guidance, and it is enabled on the Security OU during setup or landing zone updates of 3.0 and later. The second control has **Strongly recommended** guidance, and it is optionally applied to any OUs other than the Security OU, which already has the mandatory control protection enforced.

Mandatory control: Detect whether shared accounts under the Security organizational unit have AWS CloudTrail or CloudTrail Lake enabled

Strongly recommended control: Detect whether an account has AWS CloudTrail or CloudTrail Lake enabled

For more information about the new controls, see The AWS Control Tower controls library.

An update to the Region deny control

We updated the **NotAction** list in the Region deny control to include actions by some additional services, listed below:

```
"chatbot:*",
"s3:GetAccountPublic",
"s3:DeleteMultiRegionAccessPoint",
"s3:DescribeMultiRegionAccessPointOperation",
"s3:GetMultiRegionAccessPoint",
"s3:GetMultiRegionAccessPointPolicy",
"s3:GetMultiRegionAccessPointPolicyStatus",
"s3:ListMultiRegionAccessPoints",
"s3:GetStorageLensConfiguration",
"s3:GetStorageLensDashboard",
"s3:ListStorageLensConfigurations"
"s3:GetAccountPublicAccessBlock",
"s3:PutAccountPublic",
"s3:PutAccountPublicAccessBlock",
```

Video Walkthrough

This video (3:07) describes how to update your existing AWS Control Tower landing zone to version 3. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Update an Existing AWS Control Tower Landing Zone to Landing Zone 3.

The Organization page combines views of OUs and accounts

July 18, 2022

(No update required for AWS Control Tower landing zone)

The new **Organization** page in AWS Control Tower shows a hierarchical (tree) view of all organizational units (OUs) and accounts. It combines the information from the **OUs** and **Accounts** pages, which existed previously.

On the new page, you can see relationships between parent OUs and their nested OUs and accounts, You can take action on groupings of resources. You can configure the page view. For example, you can expand or collapse the hierarchical view, filter the view to see accounts or OUs only, choose to view only your enrolled accounts and registered OUs, or you can view groups of related resources. It is easier to ensure that your entire organization is updated properly.

Simplified account creation and enrollment

June 30, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now provides a simplified workflow for creating and enrolling accounts. You can create a new account or enroll an existing account in a single workflow, without navigating to the Service Catalog console. For more information, see Enroll an existing account.

AFT supports automated customization for shared AWS Control Tower accounts

May 27, 2022

(No update required for AWS Control Tower landing zone)

Account Factory for Terraform (AFT) now can programmatically customize and update any of your accounts that are managed by AWS Control Tower, including the management account, audit account, and log archive account, along with your enrolled accounts. You can centralize your account customization and update management, while protecting the security of your account configurations, because you scope the role that carries out the work.

The existing **AWSAFTExecution** role now deploys customizations in all accounts. You can set up IAM permissions with boundaries that limit the access of the **AWSAFTExecution** role according to your business and security requirements. You also can programmatically delegate the approved customization permissions in that role, for trusted users. As a best practice, we recommend that you restrict permissions to those that are necessary to deploy the required customizations.

AFT now creates the new **AWSAFTService** role to deploy AFT resources in all managed accounts, including the shared accounts and management account. Resources formerly were deployed by the **AWSAFTExecution** role.

The AWS Control Tower shared and management accounts are not provisioned through account factory, so they do not have corresponding provisioned products in AWS Service Catalog. Therefore, you are not able to update the shared and management accounts in Service Catalog.

Concurrent operations for all optional controls

May 18, 2022

(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports concurrent operations for preventive controls, as well as for detective controls.

With this new feature, any optional control now can be applied or removed concurrently, thereby improving the ease of use and performance for all optional controls. You can enable multiple optional controls without waiting for individual control operations to complete. The only restricted times are when AWS Control Tower is in the process of landing zone setup, or while extending governance to a new organization.

Supported functionality for preventive controls:

- Apply and remove different preventive controls on the same OU.
- Apply and remove different preventive controls on different OUs, concurrently.
- Apply and remove the same preventive control on multiple OUs, concurrently.
- You can apply and remove any preventive and detective controls, concurrently.

You can experience these control concurrency improvements in all released versions of AWS Control Tower.

When you apply preventive controls to nested OUs, the preventive controls affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered with AWS Control Tower. Preventive controls are implemented using Service Control Policies (SCPs), which are part of AWS Organizations. Detective controls are implemented using AWS Config rules. Guardrails remain in effect as you create new accounts or make changes to your existing accounts, and AWS

Control Tower provides a summary report of how each account conforms to your enabled policies. For a full list of available controls, see The AWS Control Tower controls library.

Existing security and logging accounts

May 16, 2022

(Available during initial setup.)

AWS Control Tower now provides the option for you to specify an existing AWS account as an AWS Control Tower security or logging account, during the initial landing zone setup process. This option eliminates the need for AWS Control Tower to create new, shared accounts. The security account, which is called the **Audit** account by default, is a restricted account that gives your security and compliance teams access to all accounts in your landing zone. The logging account, which is called the **Log Archive** account by default, works as a repository. It stores logs of API activities and resource configurations from all accounts in your landing zone.

By bringing your existing security and logging accounts, it is easier to extend AWS Control Tower governance into your existing organizations, or to move to AWS Control Tower from an alternate landing zone. The option for you to use existing accounts is displayed during the initial landing zone setup. It includes checks during the setup process, which ensure successful deployment. AWS Control Tower implements the necessary roles and controls on your existing accounts. It does not remove or merge any existing resources or data that exists in these accounts.

Limitation: If you plan to bring existing AWS accounts into AWS Control Tower as audit and log archive accounts, and if those accounts have existing AWS Config resources, you must delete the existing AWS Config resources before you can enroll the accounts into AWS Control Tower.

AWS Control Tower landing zone version 2.9

April 22, 2022

(Update required for AWS Control Tower landing zone to version 2.9. For information, see Update your landing zone)

AWS Control Tower landing zone version 2.9 updates the notification forwarder Lambda to use the Python version 3.9 runtime. This update addresses the deprecation of Python version 3.6, which is planned for July of 2022. For the latest information, see the Python deprecation page.

AWS Control Tower landing zone version 2.8

February 10, 2022

(Update required for AWS Control Tower landing zone to version 2.8. For information, see Update your landing zone)

AWS Control Tower landing zone version 2.8 adds functionality that aligns with recent updates to the AWS Foundational Security Best Practices.

In this release:

- Access logging is configured for the access log bucket in the Log Archive account, to keep track of access to the existing S3 access log bucket.
- Support for lifecycle policy is added. The access log for the existing S3 access log bucket is set to a default retention time of 10 years.
- Additionally, this release updates AWS Control Tower to use the AWS Service Linked Role (SLR) provided by AWS Config, in all managed accounts (not including the management account), so that you can set up and manage Config rules to match AWS Config best practices. Customers who do not upgrade will continue to use their existing role.
- This release streamlines the AWS Control Tower KMS configuration process for encrypting AWS Config data, and it improves the related status messaging in CloudTrail.
- The release includes an update to the Region deny control, to allow for the `route53-application-recovery` feature in `us-west-2`.
- Update: On February 15, 2022, we removed the dead letter queue for AWS Lambda functions.

Additional details:

- If you decommission your landing zone, AWS Control Tower does not remove the AWS Config service-linked role.
- If you deprovision an Account Factory account, AWS Control Tower does not remove the AWS Config service-linked role.

To update your landing zone to 2.8, navigate to the **Landing zone settings** page, select the 2.8 version, and then choose **Update**. After you update your landing zone, you must update all accounts that are governed by AWS Control Tower, as given in Configuration update management in AWS Control Tower.

January - December 2021

In 2021, AWS Control Tower released the following updates:

- Region deny capabilities
- Data residency features
- AWS Control Tower introduces Terraform account provisioning and customization
- New lifecycle event available
- AWS Control Tower enables nested OUs
- Detective control concurrency
- Two new Regions available
- Region deselection
- AWS Control Tower works with AWS Key Management Systems
- Controls renamed, functionality unchanged
- AWS Control Tower scans SCPs daily to check for drift
- Customized names for OUs and accounts
- AWS Control Tower landing zone version 2.7
- Three new AWS Regions available
- Govern selected Regions only
- AWS Control Tower now extends governance to existing OUs in your AWS organizations
- AWS Control Tower provides bulk account updates

Region deny capabilities

November 30, 2021

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now provides Region deny capabilities, which assist you in limiting access to AWS services and operations for enrolled accounts in your AWS Control Tower environment. The Region deny feature complements existing Region selection and Region deselection features in AWS Control Tower. Together, these features help you to address compliance and regulatory concerns, while balancing the costs associated with expanding into additional Regions.

For example, AWS customers in Germany can deny access to AWS services in Regions outside of the Frankfurt Region. You can select restricted Regions during the AWS Control Tower set up process, or in the **Landing zone settings** page. The Region deny feature is available when you update your AWS Control Tower landing zone version. Select AWS services are exempt from Region deny capabilities. To learn more, see Configure the Region deny control.

Data residency features

November 30, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now offers purpose-built controls to help ensure that any customer data you upload to AWS services is located only in the AWS Regions that you specify. You can select the AWS Region or Regions in which your customer data is stored and processed. For a full list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

For granular control, you can apply additional controls, such as **Disallow Amazon Virtual Private Network (VPN) connections**, or **Disallow internet access for an Amazon VPC instance**. You can view the compliance status of the controls in the AWS Control Tower console. For a full list of available controls, see The AWS Control Tower controls library.

AWS Control Tower introduces Terraform account provisioning and customization

November 29, 2021

(Optional update for AWS Control Tower landing zone)

You can now employ Terraform to provision and update customized accounts through AWS Control Tower, with *AWS Control Tower Account Factory for Terraform (AFT)*.

AFT provides a single Terraform infrastructure as code (IaC) pipeline, which provisions accounts managed by AWS Control Tower. Customizations during provisioning help to meet your business and security policies, before you give the accounts to end-users.

The AFT automated account creation pipeline monitors until account provisioning is complete, and then it continues, triggering additional Terraform modules that enhance the account with any necessary customizations. As an additional part of the customization process, you can configure the pipeline to install your own custom Terraform modules, and you can choose to add any of the AFT Feature Options, which are provided by AWS for common customizations.

Get started with AWS Control Tower Account Factory for Terraform by following the steps provided in the *AWS Control Tower User Guide*, Deploy AWS Control Tower Account Factory for Terraform (AFT) , and by downloading AFT for your Terraform instance. AFT supports Terraform Cloud, Terraform Enterprise, and Terraform Open Source distributions.

New lifecycle event available

November 18, 2021

(No update required for AWS Control Tower landing zone)

The `PrecheckOrganizationalUnit` event logs whether any resources block the **Extend governance** task from success, including resources in nested OUs. For more information, see `PrecheckOrganizationalUnit`.

AWS Control Tower enables nested OUs

November 16, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now enables you to include nested OUs as part of your landing zone.

AWS Control Tower provides support for nested organizational units (OUs), allowing you to organize accounts into multiple hierarchy levels, and to enforce preventive controls hierarchically. You can register OUs containing nested OUs, create and register OUs under parent OUs, and enable controls on any registered OU, regardless of depth. To support this functionality, the console shows the number of governed accounts and OUs.

With nested OUs, you can align your AWS Control Tower OUs to the AWS multi-account strategy, and you can reduce the time required to enable controls on multiple OUs, by enforcing controls at the parent OU level.

Key considerations

1. You can register existing, multi-level OUs with AWS Control Tower one OU at a time, starting with the top-level OU and then proceeding down the tree. For more information, see Expand from flat OU structure to nested OU structure.

2. Accounts directly under a registered OU are enrolled automatically. Accounts further down the tree can be enrolled by registering their immediate parent OU.
3. Preventive controls (SCPs) are inherited down the hierarchy automatically; SCPs applied to the parent are inherited by all nested OUs.
4. Detective controls (AWS Config rules) are NOT inherited automatically.
5. Compliance with detective controls is reported by each OU.
6. SCP drift on an OU affects all accounts and OUs under it.
7. You cannot create new nested OUs under the Security OU (Core OU).

Detective control concurrency

November 5, 2021

(Optional update for AWS Control Tower landing zone)

AWS Control Tower detective controls now support concurrent operations for detective controls, improving the ease of use and performance. You can enable multiple detective controls without waiting for individual control operations to complete.

Supported functionality:

- Enable different detective controls on the same OU (for example, **Detect Whether MFA for the Root User is Enabled** and **Detect Whether Public Write Access to Amazon S3 Buckets is Allowed**).
- Enable different detective controls on different OUs, concurrently.
- Guardrail error messaging has been improved to give additional guidance for supported control concurrency operations.

Not supported in this release:

- Enabling the same detective control on multiple OUs concurrently is not supported.
- *Preventive* control concurrency is not supported.

You can experience the detective control concurrency improvements in all versions of AWS Control Tower. It is recommended that customers not currently on version 2.7 perform a landing zone update to take advantage of other features, such as Region selection and deselection, which are available in the latest version.

Two new Regions available

July 29, 2021

(Update required for AWS Control Tower landing zone)

AWS Control Tower is now available in two additional AWS Regions: South America (Sao Paulo), and Europe (Paris). This update expands AWS Control Tower availability to 15 AWS Regions.

If you are new to AWS Control Tower, you can launch it right away in any of the supported Regions. During the launch, you can select the Regions in which you want AWS Control Tower to build and govern your multi-account environment.

If you already have an AWS Control Tower environment and you want to extend or remove AWS Control Tower governance features in one or more supported Regions, go to the **Landing Zone Settings** page in your AWS Control Tower dashboard, then select the Regions. After updating your landing zone, you must then update all accounts that are governed by AWS Control Tower.

Region deselection

July 29, 2021

(Optional update for AWS Control Tower landing zone)

AWS Control Tower Region deselection enhances your ability to manage the geographical footprint of your AWS Control Tower resources. You can deselect Regions you would no longer like AWS Control Tower to govern. This feature provides you with the capability to address compliance and regulatory concerns while balancing the costs associated with expanding into additional Regions.

Region deselection is available when you update your AWS Control Tower landing zone version.

When you use Account Factory to create a new account or enroll a pre-existing member account, or when you select **Extend Governance** to enroll accounts in a pre-existing organizational unit, AWS Control Tower deploys its governance capabilities—which include centralized logging, monitoring, and controls— in your chosen Regions in the accounts. Choosing to deselect a Region and remove AWS Control Tower governance from that Region removes that governance functionality, but it does not inhibit your users' ability to deploy AWS resources or workloads into those Regions.

AWS Control Tower works with AWS Key Management Systems

July 28, 2021

(Optional update for AWS Control Tower landing zone)

AWS Control Tower provides you the option to use an AWS Key Management Service (AWS KMS) key. A key is provided and managed by you, to secure the services that AWS Control Tower deploys, including AWS CloudTrail, AWS Config, and the associated Amazon S3 data. AWS KMS encryption is an enhanced level of encryption over the SSE-S3 encryption that AWS Control Tower uses by default.

The integration of AWS KMS support into AWS Control Tower aligns with the **AWS Foundational Security Best Practices**, which recommend an added layer of security for your sensitive log files. You should use AWS KMS–managed keys (SSE-KMS) for encryption at rest. AWS KMS encryption support is available when you set up a new landing zone or when you update your existing AWS Control Tower landing zone.

To configure this functionality, you can select **KMS Key Configuration** during your initial landing zone setup. You can choose an existing KMS key, or you can select a button that directs you to the AWS KMS console to create a new one. You also have the flexibility to change from default encryption to SSE-KMS, or to a different SSE-KMS key.

For an existing AWS Control Tower landing zone, you can perform an update to start using AWS KMS keys.

Controls renamed, functionality unchanged

July 26, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower is revising certain control names and descriptions to better reflect the policy intentions of the control. The revised names and descriptions help you understand more intuitively the ways in which controls embody the policies of your accounts. For example, we changed part of the names of detective controls from “Disallow” to “Detect” because the detective control itself does not stop a specific action, it only detects policy violations and provides alerts through the dashboard.

Control functionality, guidance, and implementation remain unchanged. Only the control names and descriptions have been revised.

AWS Control Tower scans SCPs daily to check for drift

May 11, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now performs daily automated scans of your managed SCPs to verify that the corresponding controls are applied correctly and that they have not drifted. If a scan discovers drift, you will receive a notification. AWS Control Tower sends only one notification per drift issue, so if your landing zone already is in a state of drift, you will not receive additional notifications unless a new drift item is found.

Customized names for OUs and accounts

April 16, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now allows you to customize your landing zone naming. You can retain the names that AWS Control Tower recommends for the organizational units (OUs) and core accounts, or you can modify these names during the initial landing zone set up process.

The default names that AWS Control Tower provides for the OUs and core accounts match the AWS multi-account best practices guidance. However, if your company has specific naming policies, or if you already have an existing OU or account with the same recommended name, the new OU and account naming functionality gives you the flexibility to address those constraints.

Separately from that workflow change during setup, the OU formerly known as the Core OU is now called the Security OU, and the OU formerly known as the Custom OU is now called the Sandbox OU. We made this change to improve our alignment with overall AWS best practices guidance for naming.

New customers will see these new OU names. Existing customers will continue to see the original names of these OUs. You may encounter some inconsistencies in OU naming while we are updating our documentation to the new names.

To get started with AWS Control Tower from the AWS Management Console, go to the AWS Control Tower console, and select **Set up landing zone** in the top right. For additional information, you can read about planning your AWS Control Tower landing zone.

AWS Control Tower landing zone version 2.7

April 8, 2021

(Update required for AWS Control Tower landing zone to version 2.7. For information, see Update your landing zone)

With AWS Control Tower version 2.7, AWS Control Tower introduces four new mandatory preventative Log Archive controls that implement policy solely on AWS Control Tower resources. We have adjusted the guidance on four existing Log Archive controls from mandatory to elective, because they set policy for resources outside of AWS Control Tower. This control change and expansion provides the ability to separate Log Archive governance for resources within AWS Control Tower from governance of resources outside of AWS Control Tower.

The four changed controls can be used in conjunction with the new mandatory controls to provide governance to a broader set of AWS Log Archives. Existing AWS Control Tower environments will keep these four changed controls enabled automatically, for environment consistency; however, these elective controls now can be disabled. New AWS Control Tower environments must enable all elective controls. **Existing environments must disable the formerly mandatory controls before adding encryption to Amazon S3 buckets that are not deployed by AWS Control Tower.**

New mandatory controls:

- Disallow Changes to Encryption Configuration for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Logging Configuration for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Bucket Policy for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Lifecycle Configuration for AWS Control Tower Created S3 Buckets in Log Archive

Guidance changed from Mandatory to Elective:

- Disallow Changes to Encryption Configuration for all Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive]
- Disallow Changes to Logging Configuration for all Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive]

- Disallow Changes to Bucket Policy for all Amazon S3 Buckets [Previously: Disallow Policy Changes to Log Archive]
- Disallow Changes to Lifecycle Configuration for all Amazon S3 Buckets [Previously: Set a Retention Policy for Log Archive]

AWS Control Tower version 2.7 includes changes to the AWS Control Tower landing zone blueprint that can cause incompatibility with previous versions after you upgrade to 2.7.

- In particular, AWS Control Tower version 2.7 enables `BlockPublicAccess` automatically on S3 buckets deployed by AWS Control Tower. You can turn this default off if your workload requires access across accounts. For more information about what happens with `BlockPublicaccess` enabled, see [Blocking public access to your Amazon S3 storage](#).
- AWS Control Tower version 2.7 includes a requirement for HTTPS. All requests sent to S3 buckets deployed by AWS Control Tower must use secure socket layer (SSL). Only HTTPS requests are allowed to pass. If you use HTTP (without SSL) as an endpoint to send the requests, this change gives you an access denied error, which can potentially break your workflow. **This change cannot be reverted after the 2.7 update to your landing zone.**

  We recommend that you change your requests to use TLS instead of HTTP.

Three new AWS Regions available

April 8, 2021

(Update required for AWS Control Tower landing zone)

AWS Control Tower is available in three additional AWS Regions: Asia Pacific (Tokyo) Region, Asia Pacific (Seoul) Region, and Asia Pacific (Mumbai) Region. A landing zone update to version 2.7 is required for expanding governance into these Regions.

Your landing zone is not expanded automatically into these Regions when you perform the update to version 2.7, you must view and select them in the Regions table for inclusion.

Govern selected Regions only

February 19, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower Region selection provides better ability to manage the geographical footprint of your AWS Control Tower resources. To expand the number of Regions in which you host AWS resources or workloads – for compliance, regulatory, cost, or other reasons – you can now select the additional Regions to govern.

Region selection is available when you set up a new landing zone or update your AWS Control Tower landing zone version. When you use Account Factory to create a new account or enroll a pre-existing member account, or when you use **Extend Governance** to enroll accounts in a pre-existing organizational unit, AWS Control Tower deploys its governance capabilities of centralized logging, monitoring, and controls in your chosen Regions in the accounts. For more information about selecting Regions, see Configure your AWS Control Tower Regions.

AWS Control Tower now extends governance to existing OUs in your AWS organizations

January 28, 2021

(No update required for AWS Control Tower landing zone)

Extend governance to existing organizational units (OUs) (those not in AWS Control Tower) from within the AWS Control Tower console. With this feature, you can bring top-level OUs and included accounts under AWS Control Tower governance. For information about extending governance to an entire OU, see Register an existing organizational unit with AWS Control Tower.

When you register an OU, AWS Control Tower performs a series of checks to ensure successful extension of governance and enrollment of accounts within the OU. For more information about common issues associated with the initial registration of an OU, see Common causes of failure during registration or re-registration.

You can also visit the AWS Control Tower product webpage or visit YouTube to watch this video about getting started with AWS Control Tower for AWS Organizations.

AWS Control Tower provides bulk account updates

January 28, 2021

(No update required for AWS Control Tower landing zone)

With the bulk update feature, you can now update all accounts in a registered AWS Organizations organizational unit (OU) containing up to 300 accounts, with a single click, from the AWS Control

Tower dashboard. This is particularly useful in cases where you update your AWS Control Tower landing zone and must also update your enrolled accounts to align them to the current landing zone version.

This feature also helps you keep your accounts up to date when you update your AWS Control Tower landing zone to expand to new regions, or when you want to re-register an OU to ensure that all accounts in that OU have the latest controls applied. Bulk account update eliminates the need to update one account at a time or use an external script to perform the update on multiple accounts.

For information about updating a landing zone, see Update your landing zone.

For information about registering or re-registering an OU, see Register an existing organizational unit with AWS Control Tower.

January - December 2020

In 2020, AWS Control Tower released the following updates:

- AWS Control Tower console now links to external AWS Config rules
- AWS Control Tower now available in additional Regions
- Guardrail update
- AWS Control Tower console shows more detail about OUs and accounts
- Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations
- Customizations for AWS Control Tower solution
- General availability of AWS Control Tower version 2.3
- Single-step account provisioning in AWS Control Tower
- AWS Control Tower decommissioning tool
- AWS Control Tower lifecycle event notifications

AWS Control Tower console now links to external AWS Config rules

December 29, 2020

(Update required for AWS Control Tower landing zone to version 2.6. For information, see Update your landing zone)

AWS Control Tower now includes an organization-level aggregator, which assists in detecting external AWS Config rules. This provides you with visibility in the AWS Control Tower console to see the existence of externally created AWS Config rules in addition to those AWS Config rules created by AWS Control Tower. The aggregator allows AWS Control Tower to detect external rules and provide a link to the AWS Config console without the need for AWS Control Tower to gain access to unmanaged accounts.

With this feature, you now have a consolidated view of detective controls applied to your accounts so you can track compliance and determine if you need additional controls for your account. For information, see How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts.

AWS Control Tower now available in additional Regions

November 18, 2020

(Update required for AWS Control Tower landing zone to version 2.5. For information, see Update your landing zone)

AWS Control Tower is now available in 5 additional AWS Regions:

- Asia Pacific (Singapore) Region
- Europe (Frankfurt) Region
- Europe (London) Region
- Europe (Stockholm) Region
- Canada (Central) Region

The addition of these 5 AWS Regions is the only change introduced for version 2.5 of AWS Control Tower.

AWS Control Tower is also available in US East (N. Virginia) Region, US East (Ohio) Region, US West (Oregon) Region, Europe (Ireland) Region, and Asia Pacific (Sydney) Region. With this launch AWS Control Tower is now available in 10 AWS Regions.

This landing zone update includes all Regions listed and cannot be undone. After updating your landing zone to version 2.5, you must manually update all enrolled accounts for AWS Control Tower to govern in the 10 supported AWS Regions. For information, see Configure your AWS Control Tower Regions.

Guardrail update

October 8, 2020

(No update required for AWS Control Tower landing zone)

An updated version has been released for the mandatory control `AWS-GR_IAM_ROLE_CHANGE_PROHIBITED`.

This change to the control is required because accounts that are being enrolled automatically into AWS Control Tower must have the `AWSControlTowerExecution` role enabled. The previous version of the control prevents this role from being created.

For more information, see Disallow Changes to AWS IAM Roles Set Up by AWS Control Tower and AWS CloudFormation in the AWS Control Tower Controls Reference Guide.

AWS Control Tower console shows more detail about OUs and accounts

July 22, 2020

(No update required for AWS Control Tower landing zone)

You can view your organizations and accounts that are not enrolled in AWS Control Tower, alongside organizations and accounts that are enrolled.

Within the AWS Control Tower console, you can view more detail about your AWS accounts and organizational units (OUs). The **Accounts** page now lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can now search, sort, and filter across all tables.

Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations

April 22, 2020

(No update required for AWS Control Tower landing zone)

AWS Organizations customers can now use AWS Control Tower to manage newly created organizational units (OUs) and accounts by taking advantage of these new capabilities:

- Existing AWS Organizations customers can now set up a new landing zone for new organizational units (OUs) in their existing management account. You can create new OUs in AWS Control Tower and create new accounts in those OUs with AWS Control Tower governance.
- AWS Organizations customers can enroll existing accounts using the account enrollment process or through scripting.

AWS Control Tower provides an orchestration service that uses other AWS services. It's designed for organizations with multiple accounts and teams who are looking for the easiest way to set up their new or existing multi-account AWS environment and govern at scale. With an organization governed by AWS Control Tower, cloud administrators know that accounts in the organization are compliant with established policies. Builders benefit because they can provision new AWS accounts quickly, without undue concerns about compliance.

For information about setting up a landing zone, see Plan your AWS Control Tower landing zone. You can also visit the AWS Control Tower product webpage or visit YouTube to watch this video about getting started with AWS Control Tower for AWS Organizations.

In addition to this change, the **Quick account provisioning** capability in AWS Control Tower was renamed to **Enroll account**. It now permits enrollment of existing AWS accounts as well as creation of new accounts. For more information, see Enroll an existing account.

Customizations for AWS Control Tower solution

March 17, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes a new reference implementation that makes it easy for you to apply custom templates and policies to your AWS Control Tower landing zone.

With customizations for AWS Control Tower, you can use AWS CloudFormation templates to deploy new resources to existing and new accounts within your organization. You can also apply custom service control policies (SCPs) to those accounts in addition to the SCPs already provided by AWS Control Tower. Customizations for AWS Control Tower pipeline integrate with AWS Control Tower lifecycle events and notifications (Lifecycle Events in AWS Control Tower) to ensure that resource deployments stay in sync with your landing zone.

The deployment documentation for this AWS Control Tower solution architecture is available through the AWS Solutions web page.

General availability of AWS Control Tower version 2.3

March 5, 2020

(Update required for AWS Control Tower landing zone. For information, see Update your landing zone.)

AWS Control Tower is now available in the Asia Pacific (Sydney) AWS Region, in addition to the US East (Ohio), US East (N. Virginia), US West (Oregon), and Europe (Ireland) Regions. The addition of the Asia Pacific (Sydney) Region is the only change introduced for version 2.3 of AWS Control Tower.

If you have not used AWS Control Tower previously, you can launch it today in any of the supported Regions. If you are already using AWS Control Tower and want to extend its governance features to the Asia Pacific (Sydney) Region in your accounts, go to the **Settings** page in your AWS Control Tower dashboard. From there, update your landing zone to the latest release. Then, update your accounts individually.

> **Note**
>
> Updating your landing zone does not automatically update your accounts. If you have more than a few accounts, the required updates can be time-consuming. For that reason, we recommend that you avoid expanding your AWS Control Tower landing zone into Regions in which you do not require your workloads to run.

For information about the expected behavior of detective controls as a result of a deployment to a new Region, see Configure your AWS Control Tower Regions.

Single-step account provisioning in AWS Control Tower

March 2, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports single-step account provisioning through the AWS Control Tower console. This feature allows you to provision new accounts from within the AWS Control Tower console.

To use the simplified form, navigate to **Account Factory** in the AWS Control Tower console and then choose **Quick account provisioning**. AWS Control Tower assigns the same email address to the provisioned account and to the single sign-on (IAM Identity Center) user that is created for the account. If you require these two email addresses to be different, you must provision your account through Service Catalog.

Update accounts that you create through quick account provisioning by using Service Catalog and the AWS Control Tower account factory, just like updates to any other account.

> **Note**
>
> In April 2020, the **Quick account provisioning** capability was renamed to **Enroll account**. In June 2022, the ability to create and update accounts in the AWS Control Tower console was separated from the ability to enroll AWS accounts. For more information, see Enroll an existing account.

AWS Control Tower decommissioning tool

February 28, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports an automated decommissioning tool to assist you in cleaning up resources allocated by AWS Control Tower. If you no longer intend to use AWS Control Tower for your enterprise, or if you require a major redeployment of your organizational resources, you may want to clean up the resources created when you initially set up your landing zone.

To decommission your landing zone by using a process that is mostly automated, contact AWS Support to get assistance with the additional steps that are required. For more information about decommissioning, see Decommission an AWS Control Tower landing zone.

AWS Control Tower lifecycle event notifications

January 22, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower announces the availability of lifecycle event notifications. A lifecycle event marks the completion of an AWS Control Tower action that can change the state of resources such as organizational units (OUs), accounts, and controls that are created and managed by AWS

Control Tower. Lifecycle events are recorded as AWS CloudTrail events and delivered to Amazon EventBridge as events.

AWS Control Tower records lifecycle events at the completion of the following actions that can be performed using the service: creating or updating a landing zone; creating or deleting an OU; enabling or disabling a control on an OU; and using account factory to create a new account or to move an account to another OU.

AWS Control Tower uses multiple AWS services to build and govern a best practices multi-account AWS environment. It can take several minutes for an AWS Control Tower action to complete. You can track lifecycle events in the CloudTrail logs to verify if the originating AWS Control Tower action completed successfully. You can create an EventBridge rule to notify you when CloudTrail records a lifecycle event or to automatically trigger the next step in your automation workflow.

June - December 2019

From June 24 through December 31, 2019, AWS Control Tower released the following updates:

- General availability of AWS Control Tower version 2.2
- New elective controls in AWS Control Tower
- New detective controls in AWS Control Tower
- AWS Control Tower accepts email addresses for shared accounts with different domains than the management account
- General availability of AWS Control Tower version 2.1

General availability of AWS Control Tower version 2.2

November 13, 2019

(Update required for AWS Control Tower landing zone. For information, see Update your landing zone.)

AWS Control Tower version 2.2 provides three new preventive controls that prevent drift in accounts:

- Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower
- Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower

- Disallow Deletion of Log Archive

A control is a high-level rule that provides ongoing governance for your overall AWS environment. When you create your AWS Control Tower landing zone, the landing zone and all the organizational units (OUs), accounts, and resources are compliant with the governance rules enforced by your chosen controls. As you and your organization members use the landing zone, changes (accidental or intentional) in this compliance status may occur. Drift detection helps you identify resources that need changes or configuration updates to resolve the drift. For more information, see Detect and resolve drift in AWS Control Tower.

New elective controls in AWS Control Tower

September 05, 2019

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes the following four new elective controls:

- Disallow Delete Actions on Amazon S3 Buckets Without MFA
- Disallow Changes to Replication Configuration for Amazon S3 Buckets
- Disallow Actions as a Root User
- Disallow Creation of Access Keys for the Root User

A control is a high-level rule that provides ongoing governance for your overall AWS environment. Guardrails enable you to express your policy intentions. For more information, see About controls in AWS Control Tower.

New detective controls in AWS Control Tower

August 25, 2019

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes the following eight new detective controls:

- Detect Whether Versioning for Amazon S3 Buckets is Enabled
- Detect Whether MFA is Enabled for IAM Users of the AWS Console
- Detect Whether MFA is Enabled for IAM Users

- Detect Whether Amazon EBS Optimization is Enabled for Amazon EC2 Instances
- Detect Whether Amazon EBS Volumes are Attached to Amazon EC2 Instances
- Detect Whether Public Access to Amazon RDS Database Instances is Enabled
- Detect Whether Public Access to Amazon RDS Database Snapshots is Enabled
- Detect Whether Storage Encryption is Enabled for Amazon RDS Database Instances

A control is a high-level rule that provides ongoing governance for your overall AWS environment. A detective control detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. For more information, see About controls in AWS Control Tower.

AWS Control Tower accepts email addresses for shared accounts with different domains than the management account

August 01, 2019

(No update required for AWS Control Tower landing zone)

In AWS Control Tower, you can now submit email addresses for shared accounts (log archive and audit member) and child accounts (vended using account factory) whose domains are different from the management account's email address. This feature is available only when you create a new landing zone and when you provision new child accounts.

General availability of AWS Control Tower version 2.1

June 24, 2019

(Update required for AWS Control Tower landing zone. For information, see Update Your Landing Zone.)

AWS Control Tower is now generally available and supported for production use. AWS Control Tower is intended for organizations with multiple accounts and teams who are looking for the easiest way to set up their new multi-account AWS environment and govern at scale. With AWS Control Tower, you can help make sure that accounts in your organization are compliant with established policies. End users on distributed teams can provision new AWS accounts quickly.

Using AWS Control Tower, you can set up a landing zone that employs best practices such as configuring a multi-account structure using AWS Organizations, managing user identities and federated access with AWS IAM Identity Center, enabling account provisioning through Service Catalog, and creating a centralized log archive using AWS CloudTrail and AWS Config.

For ongoing governance, you can enable pre-configured controls, which are clearly defined rules for security, operations, and compliance. Guardrails help prevent deployment of resources that don't conform to policies and continuously monitor deployed resources for nonconformance. The AWS Control Tower dashboard provides centralized visibility into an AWS environment including accounts provisioned, controls enabled, and the compliance status of accounts.

You can set up a new multi-account environment with a single click in the AWS Control Tower console. There are no additional charges or upfront commitments to use AWS Control Tower. You pay only for those AWS services that you enabled to set up a landing zone and implement selected controls.

Document history

- **Latest documentation update:** August 20, 2025

The following table describes important changes to the *AWS Control Tower User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

Change	Description	Date
AWS Control Tower supports IPv6	AWS Control Tower adds support for IPv6 endpoints.	August 18, 2025
Account Factory for Terraform version 1.15.0	AFT version 1.15.0 is available .	July 28, 2025
Updated controls with Nitro instance types	Updated eight proactive controls to add new instance types.	July 24, 2025
AWS Control Tower available in Asia Pacific (Taipei)	AWS Control Tower adds support for Asia Pacific (Taipei).	July 23, 2025
AWS Control Tower supports AWS PrivateLink	AWS Control Tower adds support for AWS PrivateLink.	June 30, 2025
Service-linked AWS Config controls	AWS Control Tower deploys service-linked AWS Config controls.	June 12, 2025
Enhanced metadata. additional controls, new frameworks	AWS Control Tower introduce s additional integrated controls from AWS Config, with enhanced metadata for all controls in the Control Catalog, and support for more industry frameworks..	June 12, 2025

Enabled controls view	AWS Control Tower introduces a comprehensive console view of enabled controls.	May 21, 2025
AFT supports new configurations	AWS Control Tower AFT supports three additional optional configurations at deployment.	May 13, 2025
Enhanced baselines	AWS Control Tower now reports baseline drift.	May 12, 2025
AWS Control Tower available in AWS Asia Pacific (Thailand) and Mexico (Central) Regions	The following Regions are available: Asia Pacific (Thailand) and Mexico (Central).	May 9, 2025
AWS Control Tower adds 233 AWS Config controls	Added new Config controls, revamped metadata tables, Region tables, Global identifier page.	April 11, 2025
AWS Control Tower updates a service-linked role	Updates to `AWSControlTowerAccountServiceRolePolicy` .	December 10, 2024
AWS Control Tower CfCT supports GitHub	A new option for a third-party configuration source.	December 9, 2024
AWS Control Tower preventive controls with declarative policies	A new type of policy implements a new type of preventive control.	December 1, 2024
AWS Control Tower integrates with AWS Backup	You can set up plans to back up your AWS Control Tower resources.	November 25, 2024
AWS Control Tower integrates AWS Config controls	AWS Control Tower integrates selected AWS Config controls.	November 21, 2024

AWS Control Tower improves hook management	AWS Control Tower now manages hooks for proactive controls.	November 20, 2024
Control policy drift reported	AWS Control Tower reports a new type of drift.	November 15, 2024
AWS Control Tower launches managed resource control policies	A new type of preventive controls, implemented with RCPs.	November 15, 2024
AWS Control Tower adds ResetEnabledControl API	A new API for managing control drift.	November 14, 2024
Updated `GetControl` API	Two new control fields for `GetControl` .	November 8, 2024
AWS Control Tower AFT supports Gitlab	A new option for a third-party configuration source.	October 23, 2024
AWS Control Tower available in AWS Asia Pacific (Malaysia) Region	A new Region is available, Malaysia (Kuala Lumpur).	October 21, 2024
AWS Control Tower supports up to 1000 accounts per OU	An increase of the limit on accounts per OU.	August 30, 2024
AWS Control Tower adds landing zone version selection	Update or repair your landing zone without moving to the latest version, if you run 3.1 or newer.	August 15, 2024
GetControl and ListControls API operations available	Two new Control Catalog operations help you find more information about controls.	August 6, 2024
AWS Control Tower supports AFT and CfCT in opt-in Regions	AFT and CfCT are available in additional AWS Regions.	July 18, 2024

AWS Control Tower adds the `ListLandingZoneOperations` API	A new API that allows you to retrieve recent operations for your landing zone.	June 26, 2024
AWS Control Tower supports up to 100 concurrent control operations	An increase of the concurrent control operations quota to 100.	May 20, 2024
AWS Control Tower available in AWS Calgary West (Canada) Region	AWS Control Tower is available in Canada West (Calgary) Region.	May 3, 2024
AWS Control Tower supports self-service quota adjustments	AWS Control Tower is integrated with AWS Service Quotas in the console.	April 25, 2024
Moved documentation for controls to a new guide	AWS Control Tower published the *Controls Reference Guide*.	April 21, 2024
Tagging `EnabledControl` resources in AWS CloudFormation	AWS Control Tower supports adding tags to `EnabledControl` resources, by means of AWS CloudFormation templates.	February 22, 2024
Baseline APIs available	AWS Control Tower released new APIs for registering OUs programmatically.	February 14, 2024
AWS Control Tower landing zone version 3.3	AWS Control Tower landing zone version 3.3 available.	December 14, 2023
AWS Control Tower announces controls to assist digital sovereignty	AWS Control Tower released a group of controls to help customers with digital sovereignty requirements.	November 27, 2023

AWS Control Tower supports landing zone APIs	AWS Control Tower supports configuring and launching landing zones using new APIs.	November 26, 2023
AWS Control Tower supports tagging enabled controls	AWS Control Tower supports tagging enabled controls, in console and with new APIs.	November 10, 2023
AWS Control Tower available in Asia Pacific (Melbourne) AWS Region	Available in Asia Pacific (Melbourne) Region.	November 3, 2023
New control API available	AWS Control Tower released a new control API.	October 14, 2023
AWS Control Tower launches new controls	AWS Control Tower released new proactive and detective controls.	October 5, 2023
AWS Control Tower reports drift from disabling trusted access	AWS Control Tower notifies customers when drift occurs, if customers turn off trusted access to AWS Control Tower in AWS Organizations.	September 21, 2023
AWS Control Tower available in four additional AWS Regions	Available in Asia Pacific (Hyderabad), Europe (Spain and Zurich), and Middle East (UAE).	September 13, 2023
AWS Control Tower available in Tel Aviv Region	AWS Control Tower is available in the Tel Aviv Region, il-central-1.	August 28, 2023
AWS Control Tower launches 28 new proactive controls	AWS Control Tower released 28 new proactive controls.	July 24, 2023

AWS Control Tower deprecates 2 controls	AWS Control Tower will remove two controls from the controls library, effective August 18, 2023.	July 18, 2023
AWS Control Tower landing zone 3.2 available	AWS Control Tower landing zone version 3.2 is available.	June 16, 2023
AWS Control Tower handles accounts based on ID	AWS Control Tower tracks the AWS account ID, rather than the account's email address.	June 14, 2023
Additional Security Hub detective controls available	AWS Control Tower adds ten new controls to the controls library, for the Security Hub Service-Managed Standard: AWS Control Tower.	June 12, 2023
AWS Control Tower publishes control metadata tables	AWS Control Tower now provides tables of control metadata as part of the published documentation.	June 7, 2023
Terraform support for Account Factory Customization	Single-region support for Terraform open source blueprints in AFC.	June 6, 2023
AWS IAM self-management available for landing zone	AWS Control Tower now supports customers in choosing their identity provider for a landing zone.	June 6, 2023

New role added	AWS Control Tower added a new service-linked role, **AWSServiceRoleForAWSControlTower**, and associated policy, **AWSControlTowerAccountServiceRolePolicy**.	June 1, 2023
Mixed governance update	Update to advise customers regarding mixed governance.	June 1, 2023
Additional proactive controls available	New proactive controls assist you in governing your multi-account environment and meeting specific control objectives.	May 19, 2023
Seven additional Regions available	AWS Control Tower is now available in seven additional AWS Regions: Northern California (San Francisco), Asia Pacific (Hong Kong, Jakarta, and Osaka), Europe (Milan), Middle East (Bahrain), and Africa (Cape Town).	April 19, 2023
Change to a managed policy	We changed the **AWSControlTowerServiceRolePolicy** so that AWS Control Tower can call the `EnableRegion` , `ListRegions` , `GetRegionOptStatus` APIs that are implemented by the AWS Account Management service.	April 6, 2023

Account customization request tracing generally available	AWS Control Tower now supports the ability to trace account customization requests using the Account Factory for Terraform (AFT) workflow.	February 16, 2023
IAM best practices update	Updated guide to align with the IAM best practices recommendations. For more information, see Security best practices in IAM.	February 15, 2023
AWS Control Tower landing zone 3.1 available	AWS Control Tower landing zone 3.1 is available.	February 9, 2023
Proactive controls generally available	Proactive controls are launched from preview status to general availability.	January 24, 2023
Concurrent account operations	AWS Control Tower now supports up to five (5) concurrent actions in account factory. You can create, update, or enroll up to five accounts at a time.	December 16, 2022
Proactive controls assist in resource provisioning	AWS Control Tower now supports proactive controls, implemented through AWS CloudFormation hooks.	November 28, 2022
Account factory customization available	AWS Control Tower now supports account provisioning with customizable account templates, called blueprints, directly from the AWS Control Tower console.	November 28, 2022

Compliance status viewable for all AWS Config rules	AWS Control Tower now displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower.	November 18, 2022
Change to a managed policy	We changed the **AWSControlTowerServiceRolePolicy** so that AWS Control Tower can assume the `AWSControlTowerBlueprintAccess` role, which is needed for Account Factory customizations.	October 28, 2022
APIs for controls, AWS CloudFormation resource	AWS Control Tower now supports activation and deactivation of controls through a set of API calls, and a new AWS CloudFormation resource.	September 1, 2022
CfCT supports stack set deletion	CfCT supports stack set deletion, by setting a parameter in the manifest file.	August 26, 2022
Customized log retention	You can customize the retention policy for Amazon S3 buckets that store your AWS Control Tower CloudTrail logs, in increments of days or years, up to a maximum of 15 years.	August 15, 2022

Role drift repair available	AWS Control Tower supports repair for role drift, without a full repair of the landing zone.	August 11, 2022
Version 3.0 available	AWS Control Tower landing zone version 3.0 changes from account-based AWS CloudTrail trails to organizat ion-based trails, and it updates the managed policy to enable organization-level trails. It enables you to aggregate AWS Config information in your home Region only. Version 3.0 also includes an update to the Region deny control, and two new detective controls.	July 29, 2022
The Organization page combines views of OUs and accounts	The new **Organization** page in AWS Control Tower shows a hierarchical view of all Organizational units (OUs) and accounts.	July 18, 2022
Change to a managed policy	We changed the **AWSControlTowerServiceRolePolicy** so that customers can have organization-level AWS CloudTrail trails to aggregate AWS CloudTrail logs.	June 20, 2022

Easier enroll and update for member accounts	AWS Control Tower now gives you the capability to to enroll and update member accounts individually, from within your landing zone. Each account shows when it is available for an update. We separated the **Enroll account** button from the **Create** account workflow in Account Factory.	May 31, 2022
AFT supports customization for shared accounts	AWS Control Tower Account Factory for Terraform now supports customization for the AWS Control Tower management account, log archive, and audit accounts.	May 27, 2022
Concurrent operations for all optional controls	AWS Control Tower now allows you to apply and remove optional preventive guardails concurrently, as well as detective controls.	May 18, 2022
Existing security and logging accounts	AWS Control Tower now supports the ability to bring existing security and logging accounts, rather than creating new ones during landing zone setup.	May 16, 2022
Version 2.9 available	AWS Control Tower landing zone version 2.9 updates the notification forwarder Lambda to use the Python version 3.9 runtime.	April 22, 2022

Updated support for AWS best practices, version 2.8 available	AWS Control Tower landing zone version 2.8 provides additional support to ensure that your workloads and AWS accounts are in alignment with AWS best practices.	February 10, 2022
Region deny control	AWS Control Tower now includes a control that helps you restrict access to AWS Regions, to address complianc e and regulatory concerns.	November 30, 2021
Data residency controls	AWS Control Tower now support controls that help you manage data residency with granular control.	November 30, 2021
AWS Control Tower Account factory for Terraform	AWS Control Tower now supports Terraform for automated account provision ing and updating.	November 29, 2021
New lifecycle event available	The `PrecheckOrganizati onalUnit` event logs whether any resources block the **Extend governance** task from success, including resources in nested OUs.	November 18, 2021
Nested OUs available	AWS Control Tower now enables your landing zone to contain nested OU structures.	November 16, 2021
Detective control concurrency	AWS Control Tower detective controls now support concurrent enable and disable operations.	November 5, 2021

Two new regions available	AWS Control Tower is now available in two new AWS Regions, Europe (Paris) Region and South America (São Paulo) Region.	July 29, 2021
Region deselection	You can deselect AWS Regions that you no longer wish to govern through AWS Control Tower.	July 29, 2021
KMS keys available	You can optionally create or choose KMS keys that you manage, to encrypt your data and resources.	July 28, 2021
Change to a managed policy	We changed the **AWSControlTowerServiceRolePolicy** so that customers can use their own KMS encryption keys for AWS CloudTrail logs.	July 28, 2021
Control names changed, functionality unchanged	Certain control names and descriptions were updated to better reflect the policy intentions of the control, with no change in functionality.	July 26, 2021
Automated scans of managed SCPs	AWS Control Tower performs daily automated scans of managed SCPs to check for drift.	May 11, 2021

Customized names for OUs and accounts	AWS Control Tower allows you to provide customized names during the landing zone setup process, for essential OUs and accounts, without creating drift.	April 16, 2021
Decommissioning a landing zone is self-service	AWS Control Tower now allows you to decommiss ion a landing zone without contacting AWS Support. Decommissioning is a semi-automated process that cannot be undone. It is not the same as deleting all AWS Control Tower resources manually.	April 9, 2021
Three additional Regions	AWS Control Tower is now available in three additiona l AWS Regions: Asia Pacific (Tokyo) Region, Asia Pacific (Seoul) Region, and Asia Pacific (Mumbai) Region.	April 8, 2021

New Log Archive controls, landing zone version 2.7 available	Four new Log Archive controls provide Log Archive governance over AWS Control Tower resources, separately from governance of resources outside of AWS Control Tower. Guidance on four existing controls has changed from mandatory to elective. Version 2.7 of the AWS Control Tower landing zone includes a requirement for HTTPS, which cannot be undone after you update.	April 8, 2021
Region selection	AWS Control Tower Region selection provides better ability to manage the geographical footprint of your AWS Control Tower resources . To expand the number of Regions in which you host AWS resources or workloads – for compliance, regulatory, cost, or other reasons – you can now select the additional Regions to govern.	February 19, 2021
Register an OU and govern all of its accounts with AWS Control Tower at one time	AWS Control Tower adds the capability to register an OU, which is a way to bring multiple accounts into governance at the same time.	January 28, 2021

Multiple account updates in registered OUs	You can now update all accounts in any registered AWS Organizations organizational unit (OU) containing up to 300 accounts, with a single click, from the AWS Control Tower dashboard. The multiple account update feature, also referred to as bulk update, eliminates the need to update one account at a time, or to use an external script to perform the update on multiple accounts together.	January 28, 2021
New role for aggregating unmanaged OUs and accounts	A new role assists in detecting external AWS Config rules, so AWS Control Tower does not need to gain access to unmanaged accounts.	December 29, 2020

AWS Control Tower is available in more AWS Regions.	AWS Control Tower is now available to be deployed in the Asia Pacific (Singapore) Region, Europe (Frankfurt) Region, Europe (London) Region, Europe (Stockholm) Region, and Canada (Central) Region. With this launch AWS Control Tower is now available in 10 AWS Regions. This landing zone update includes all Regions listed, and it cannot be undone. After updating your landing zone to version 2.5, you must manually update all enrolled accounts for AWS Control Tower to govern in the 10 supported AWS Regions.	November 18, 2020
Control update	An updated version has been released for the mandatory control `AWS-GR_IAM_ROLE_CHANGE_PROHIBITED` . The updated control allows easier automated enrollment of accounts.	October 8, 2020
Related information page is now available for AWS Control Tower	The related information page makes it easier to find common tasks that may be helpful after setting up your AWS Control Tower landing zone.	September 18, 2020

AWS Control Tower console shows more detail about OUs and accounts.	Within the AWS Control Tower console, you can view more detail about your AWS accounts and organizational units (OUs). The 'Accounts' page now lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can now search, sort, and filter across all tables.	July 22, 2020
AWS Control Tower allows existing organizations to set up a landing zone	You can now launch a landing zone for AWS Control Tower in an existing organizat ion, to bring the organizat ion into governance. The **Quick account provisioning** capability in AWS Control Tower was renamed to **Enroll account** and it now permits enrollment of existing AWS accounts as well as creation of new accounts.	April 16, 2020
AWS Control Tower is now available in Asia Pacific	AWS Control Tower is now available to be deployed in the Asia Pacific (Sydney) AWS Region. This release requires manual updates to vended accounts, update only if you plan to run workloads in Asia Pacific (Sydney).	March 3, 2020

Decommissioning an AWS Control Tower landing zone is possible	AWS Support can help you permanently decommission a landing zone through a mostly automated process that preserves your organizat ions, although some manual cleanup is required.	February 27, 2020
Quick account provisioning is available in AWS Control Tower	Quick account provisioning makes it easier to launch new member accounts when your landing zone is up to date, with the **Enroll account** feature.	February 20, 2020
Lifecycle events are tracked in AWS Control Tower	Lifecycle events provide additional details for certain AWS Control Tower events, to make some workflow automation easier.	December 12, 2019
Settings and Activities pages are available for AWS Control Tower	The Settings and Activitie s pages make it easier to update your landing zone and to view logged events.	November 30, 2019
Additional preventive controls are available for AWS Control Tower	Preventive controls in AWS Control Tower keep your organization and resources aligned with your environme nt.	September 6, 2019
Additional detective controls are available for AWS Control Tower	Detective controls in AWS Control Tower give informati on about the state of your organization and resources.	August 27, 2019

AWS Control Tower is now generally available	AWS Control Tower is a service that offers the easiest way to set up and govern your multi-account AWS environment at scale.	June 24, 2019

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.