



A Guide to AWS Control Tower

Build Multi-Account AWS Architectures

Introduction

If you're planning a large-scale AWS deployment, you're probably wondering how to orchestrate multiple applications and teams on AWS. How do you make sure that every team can access AWS without your accounts turning into sprawling, ungoverned chaos?

For many companies, a multi-account structure can help meet the unique needs of each application team or business group. AWS provides free native tools like AWS Organizations to help provide central orchestration of multiple accounts, so that you can enforce security and billing configurations while still giving each team some degree of autonomy over their account.

Still, maintaining multiple AWS accounts can require a lot of annoying administrative setup and is prone to configuration drift. In 2018, AWS launched a series of new services to make that easier. AWS Control Tower is essentially an opinionated architecture that builds out a multi-account architecture with pre-configured security and access settings.

Since its launch, AWS Control Tower has become the go-to solution for Logicworks when we're doing a large-scale AWS migration. This is especially true when working with companies that have significant regulatory requirements.

In this guide, we'll address the following points:

1. **Why Multi-Account?**
2. **What is AWS Control Tower?**
3. **AWS Control Tower Best Practices**
4. **Manage Costs with Control Tower**
5. **Launching Control Tower in the Real World**

Why Multi-Account?



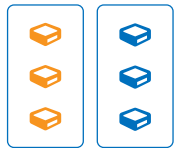
Network isolation. Ensure that services of one account are not affected by the others. By separating applications or teams into completely separate accounts, there's a better chance that an issue in one account won't affect all accounts.



Separation of concerns & modularity. An architecture that is separated into distinct services allows you to make changes, without affecting the rest of the company's accounts. It often takes less time and coding to make a change to modular infrastructure than to monolithic infrastructure where features are mixed up together.



Scalability. Need to spin up or down a new application or SDLC tier? You can do so knowing that the additional account is connected to the Hub and central security requirements.



Compliance. Limit the scope of your audits (and cut audit expenses) by maintaining regulated data in a limited number of accounts and by putting non-regulated data into another account. Also, it is often a compliance requirement to separate development and production environments (ex. SOC1 + 2). The multi-account model allows you to do this without duplicating security controls for each account.

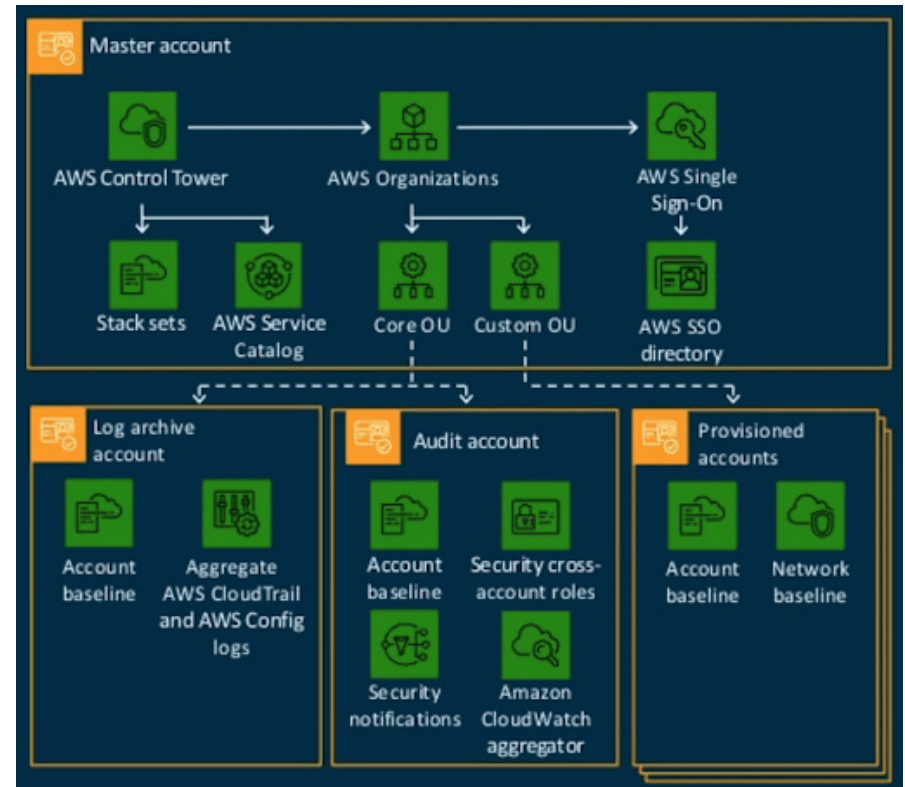
What is AWS Control Tower?

AWS Control Tower is a solution that helps automate the process of setting up and configuring multiple accounts. (Formerly known as AWS Landing Zone.) Best practices for a multi-account architecture are embedded in the solution, making AWS Control Tower perfect for companies with complex workloads and larger teams that want to quickly migrate to AWS.

Control Tower is deeply tied into AWS Organizations, a service that allows you to enroll any number of “child” accounts under a parent account and apply policies across all accounts from a single location. This extends similar functions originally used for Consolidated Billing and provides additional capabilities like AWS CloudFormation “stacksets”. Stacksets allow you to provision infrastructure across child accounts.

To start, you might have one account that has the majority of workloads. From this foundation, you can launch individual

accounts for applications, environments, business groups, or corporate entities, while keeping them separate from base infrastructure accounts.



AWS Control Tower deployment. Source: AWS

Why separate central functions from application accounts?

As Control Tower is built on the backbone of AWS Organizations, which allows you to automatically control access and permissions for child accounts. AWS Organizations allows you to define Service Control Policies to limit the services that are available to different accounts within the Organization. You can enforce policies on users of an account and define cross-account permissions to ensure your organization has the guardrails in place to maintain a secure environment. This is particularly useful for setting restrictions to powerful roles in child accounts. If the master account denies a privilege, a child account has no ability to override that restriction. Without the controls available inside an AWS Organizations structure, granting select administrative access is more difficult.

This can be a core function of your security and cost management strategies. Even if a malicious actor accesses one account, there is no way for them to access other accounts, and they may have limited privileges within that account. This limits the blast radius of certain activities.

Additionally, by having a cross-account destination for all of your logs, backups and other items you need to archive, you can more easily restrict access to those archives and ensure nothing gets deleted.

AWS Control Tower and AWS Organizations are most compelling for companies with many different IT roles who have different needs. It is also useful if you want to segregate compliance standards but still want default functionality across environments.

Why separate central functions from application accounts?

1. Core Organizational Unit with 3 accounts:

- **Master Account** - Provides the ability to create and financially manage member accounts. Also used for Account Factory provisioning and accounts, managing Organizational Units, and guardrails
- **Log Archive Account** - Contains central Amazon S3 bucket for storing logs of API activities and resource configurations from all accounts in the solution.
- **Audit Account** - A restricted account that's designed to give security and compliance teams read/write access to all accounts in the 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.

2. Within each account, an initial security baseline that includes:

- **AWS CloudTrail**, sent to a centrally managed S3 bucket in the Logging Account
- **AWS Config**, also sent to a centrally managed S3 bucket in the Logging Account
- **AWS Config** Rules enabled for monitoring encryption, IAM password policies, MFA, and security group rules
- **AWS IAM roles**, potentially including restrictions applied from the master account
- An initial **Amazon VPC network**

3. An Account Factory – essentially, an AWS Service Catalog product that allows you to automatically create new “child” accounts to the existing Organization that maintain all predefined security baselines

4. The Control Tower Dashboard – limited UI to the base Control Tower constructs. Only components deployed and managed by Control Tower are seen in the dashboard.

Control Tower can additionally work with functionality not yet exposed in the Control Tower dashboard interface, but available in the direct configuration of the foundational services. An example

of this is repointing AWS SSO to another identity provider directory, including Azure Active Directory (AD) or AWS Managed Active Directory. This AWS SSO configuration works in a Control Tower environment, but is not yet displayed in the Control Tower dashboard itself. Control Tower can also be extended with customizations or “add-ons”.

The following diagram shows the Control Tower dashboard, with a few accounts provisioned.

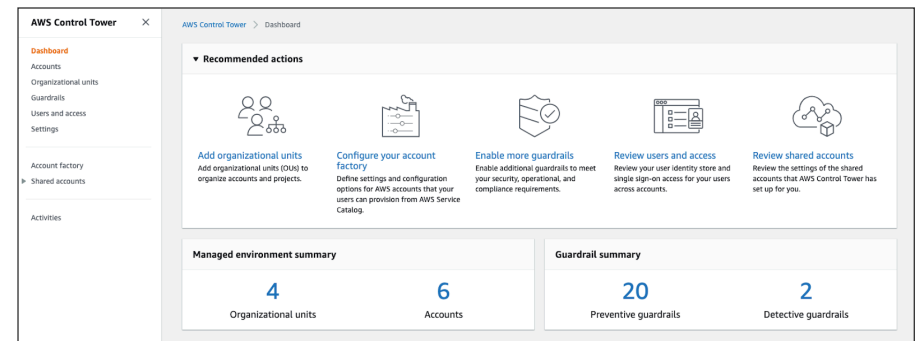


Figure 2. Control Tower Dashboard

AWS Control Tower Best Practices

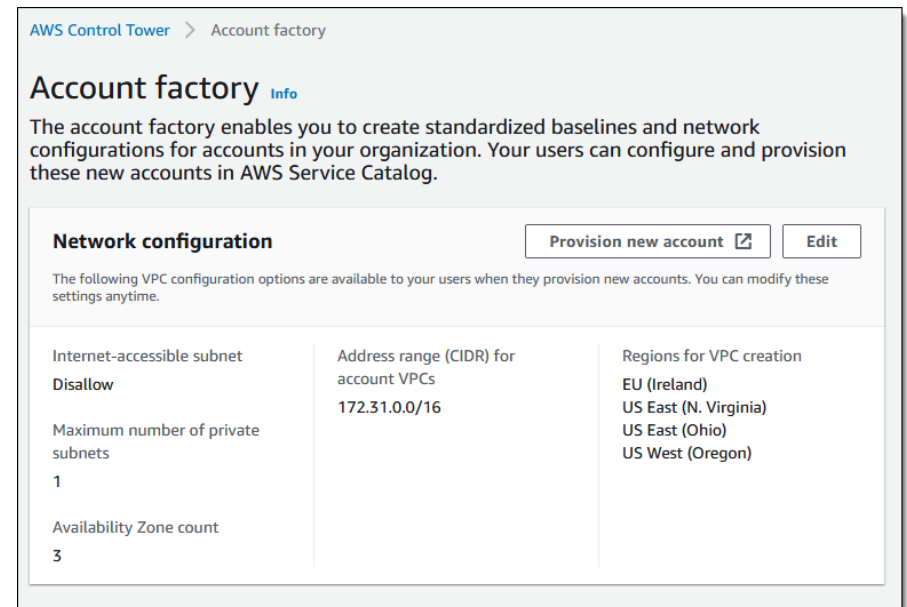
At Logicworks, we've built many Control Tower deployments for companies in a wide variety of industries. The following best practices were gleaned through trial and error with Control Tower services, and we hope they help you answer common questions.

Configure Account Factory's VPC creation

Control Tower's **Account Factory** provides an interface-driven method to create new AWS accounts that will be managed by Control Tower. These accounts will be deployed into the Control Tower's "Custom" AWS Organizational unit (OU) and will have the defined Guardrails applied.

Account Factory can also automatically create an initial VPC via "quick account provisioning". However, this initial VPC may only work with specific use cases, as it assumes all initial VPCs

will have the same private network address range (CIDR). This functionality is ideal for SaaS providers where each end customer (tenant) has their own AWS account, and the SaaS infrastructure is the same deployment in each account, with no inter-account private network connectivity.



The screenshot shows the AWS Control Tower interface for the Account Factory. The page title is "Account factory" with an "Info" link. Below the title is a description: "The account factory enables you to create standardized baselines and network configurations for accounts in your organization. Your users can configure and provision these new accounts in AWS Service Catalog." There are two buttons: "Provision new account" and "Edit".

The "Network configuration" section is highlighted. It contains the following settings:

Internet-accessible subnet	Address range (CIDR) for account VPCs	Regions for VPC creation
Disallow	172.31.0.0/16	EU (Ireland) US East (N. Virginia) US East (Ohio) US West (Oregon)
Maximum number of private subnets		
1		
Availability Zone count		
3		

If your use case requires VPCs in each account to communicate to other VPCs in the Control Tower, we recommend "disabling" the VPC process in the Configure Account Factory page and using an

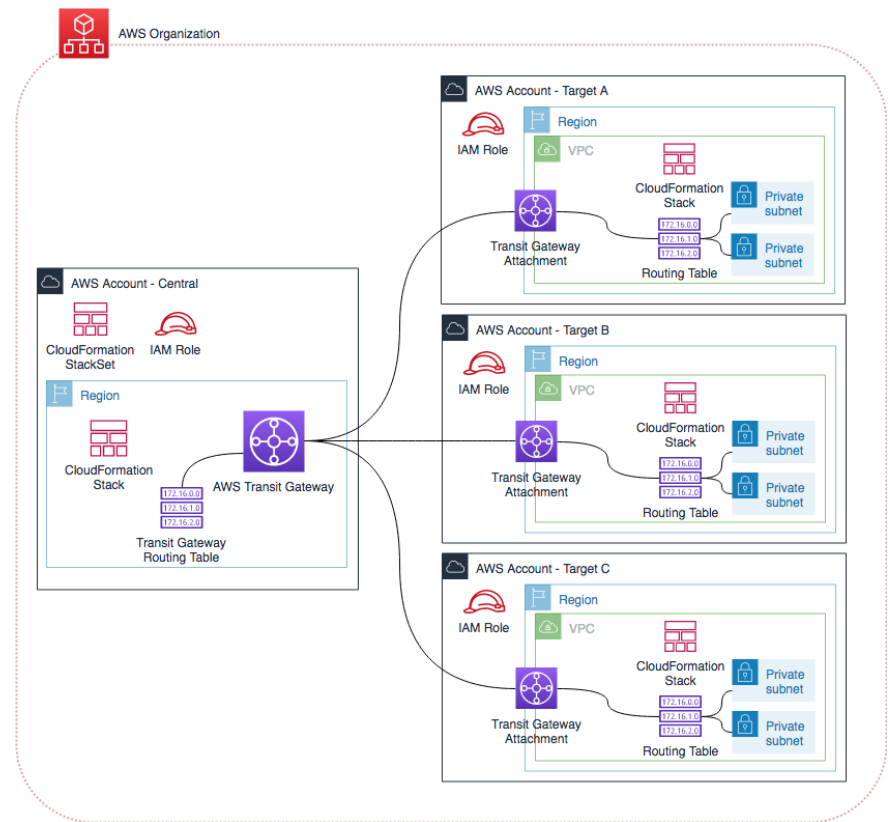
alternate method to create the VPC via automation outside of the Account Factory process. AWS has published the steps to disable automate VPC creation in the [User Guide](#).

AWS Service Catalog is another solution for creating VPCs outside of Account Factory. You can create a standardized VPC deployment Service Catalog Portfolio and Product in the Master account and share that Portfolio with the rest of the accounts for standardized VPC creation. The VPC Product can also contain adjunct resources for automated Transit Gateway attachment, Routing tables, standard NACL ruleset, Route53 Resolver shared rules attachment, etc.

Use AWS Transit Gateway

AWS **Transit Gateway** is a complimentary service with AWS Control Tower's multi-account architecture. With Transit Gateway, you are able to deploy a virtual "Centralized Router" to establish private network connectivity between the VPCs in all the accounts in your Control Tower. Transit Gateway also enables hybrid connectivity with the Control Tower by

attaching provisioned Direct Connects and Site-to-Site VPNs to the Transit Gateway. Transit Gateway can then be administered with default or segmented Transit Gateway Route Tables to enable routing between VPCs and your on-premises network or more narrowly defined segments of the network as appropriate.



In order to make Transit Gateway available for all the VPCs in the Control Tower solution, it's necessary to first share the Transit Gateway via AWS **Resource Allocation Manager** (RAM). Once the Transit Gateway is shared to the Control Tower accounts via the Organization ID, the VPCs in each account will be able to view the Transit Gateway and submit Transit Gateway attach requests to connect the VPC to the centralized Transit Gateway via software defined networking. Local VPC Route Table routes will need to be added to support various use cases like delivering all traffic to the Transit Gateway for centralized egress routing (default route 0.0.0.0/0) or route only on-premises-directed traffic to the Transit Gateway.

Enable Self-Service with AWS Service Catalog

A common design goal is to enable developer teams and business unit owners to self-provision new environments, while ensuring mandatory security tooling is in place. AWS Control Tower supports self-service provisioning of new accounts with the Account Factory. The process ensures automated deployment of security rules via Control Tower Guardrails.

Logicworks has frequently extended self-provisioning capabilities in landing zone projects by leveraging AWS Service Catalog to allow users to deploy templated default infrastructure, like default VPCs that automatically attach to Transit Gateway.

By creating a selection of default application infrastructure in Service Catalog and automatically sharing with new accounts, business unit owners of the new accounts can quickly launch approved infrastructure patterns that align with standards for security and logging in the Control Tower.

Use AWS Single Sign-On

Working with multiple AWS accounts across several teams and business units drives the need for a centralized approach for access and authentication to your AWS resources. Control Tower includes AWS **Single Sign-On** (SSO) by default as a mechanism for identity management, using a default directory. AWS SSO provides the ability to access both the AWS Management Console and programmatic access to the API via the CLI. There are two general options for identity providers with SSO:

Option 1: Use default SSO Directory

In a standard Control Tower deployment, SSO users are to be administered in the default SSO directory which is set up during the initial Control Tower deployment. The deployment also sets

up a default set of AWS SSO Permission Sets and User Groups. The Control Tower dashboard allows you to view these Permission Sets and Groups, along with a link to create Users and add them to these Groups directly via the AWS SSO portal.

AWS Control Tower ×

AWS Control Tower > Users and access

Users and access [Info](#)

Your landing zone is set up with a directory to manage user identities and single sign-on to provide your users with federated access across accounts. It offers preconfigured user groups and permission sets for you to easily manage specialized roles within your organization.

Info AWS Single Sign-On (AWS SSO) is your default directory and single sign-on. Your admin user credentials for AWS SSO have been set up and emailed to you. ×

Federated access management [View in AWS Single Sign-On](#)

Single sign-on for federated access to your users across accounts.

Access type	User portal URL
AWS Single Sign-on	https://d-90670a0022.awsapps.com/start

▶ **Permission sets**

User identity management [View in AWS Single Sign-On](#)

Your directory for managing user identities

Directory type	Directory ID
AWS SSO directory	d-90670a0022 View in AWS Single Sign-On

▶ **User groups**

Option 2: Use external identity provider

A common requirement is to integrate AWS SSO with a different identity provider. AWS SSO supports using Microsoft Active Directory (AD) or Azure AD as the user identity source. Microsoft Active Directory may be in the form of on-instance AD, or on-prem AD, or AWS Managed Microsoft AD. Depending on the specifics of AD deployment, AWS Directory Service AD Connector may also need to be deployed to proxy the authentication requests between AWS SSO and Active Directory.

Please note: re-pointing AWS SSO in Control Tower to an external provider is not currently supported in the Control Tower dashboard. However, it is a known configuration and is likely to be supported by the dashboard interface in future releases.

Reflect Internal Organization Structures and Patterns in Control Tower

A common driver for multiple accounts is that individual business unit owners or development groups are requesting

individual AWS accounts to run “their stuff”. These separate accounts simplify billing and access control for the leaders of these groups. Control Tower allows you to create these separate environments and maintain default security standards via the Control Tower Guardrails. Additionally, by using Service Control Policies, it’s possible to have some accounts like developer accounts with very open sandbox type permissions, and production accounts with more restrictive permissions.

Other Control Tower Considerations:

- Control Tower has no AWS API access
- Not all AWS Regions support Control Tower at this time (currently only 5)
- Currently Control Tower only deletes (purges) the default VPC from managed accounts in supported AWS regions
- Currently Control Tower only deploys detective Guardrails into managed accounts in supported AWS regions
- Control Tower does support in-place upgrades to new Control Tower versions, and will prompt you to upgrade via the dashboard
- Control Tower does provide some drift detection and remediation for Control Tower managed configurations

Manage Costs with Control Tower

The same basic principles of managing cloud costs in AWS apply to managing costs in Control Tower, like reducing unused resources and leveraging Spot and Reserved Instances. In addition, Control Tower provides new capabilities and some enhanced options for cost control.

Consolidated Billing

Through AWS Organizations, the billing data for all accounts is centralized into a consolidated billing report. This allows organization-wide visibility through AWS Cost Explorer and forecasting and alerting through AWS Budgets.

We recommend the following best practices and products to make the most out of this reporting:

- While it may be possible to allocate gross costs to budget owners or business units at the account level, it is still recommended to implement an organization-wide tagging standard and measure compliance through the use of AWS Config (specifically, the ["required-tags"](#) AWS Config

Managed Rule). Make these tags part of your AWS Service Catalog products. Consider automatically terminating non-compliant resources in non-production accounts.

- In addition to [traditional informative or Cost Allocation tags](#), consider more creative use of tag keys such as "expiration" or "deleteafter" to make sure resources are reviewed periodically to make sure they are still justified. Consider automating the reaping of expired resources or Service Catalog products in non-production accounts.
- Implement [Cost and Usage Reports](#), and import the data into native AWS Services such as [Amazon Athena](#), [Amazon Redshift](#), or [Amazon QuickSight](#) for more advanced query capabilities, or utilize a third party Cost Optimization/Cloud Management platform.

AWS Reserved Instances, Savings Plans, and Service Control Policies

There are more options than ever for attaining savings through the use of spending commitments via [Reserved Instances](#) and [Savings Plans](#). Through AWS Organizations, AWS Control Tower helps manage commitments in aggregate.

- By default, Savings Plan and Reserved Instance commitments apply first to the account they are purchased in, but underutilized commitments can apply to other accounts in the Organization. Consider leveraging Service Control Policies to whitelist regions, instance types or families in order to better leverage Savings Plan or Reserved Instances across accounts. Don't disable Savings Plan or Reserved Instance sharing unless absolutely necessary.
- Take advantage of lifecycle metadata (i.e., "expiration" or "deleteafter" tags) when planning your Savings Plans or Reserved Instance purchasing.

Recommendation options

Savings Plans type <input checked="" type="radio"/> Compute <input type="radio"/> EC2 Instance	Savings Plans term <input type="radio"/> 1-year <input checked="" type="radio"/> 3-year	Payment option <input checked="" type="radio"/> All upfront <input type="radio"/> Partial upfront <input type="radio"/> No upfront	Based on the past <input type="radio"/> 7 days <input type="radio"/> 30 days <input checked="" type="radio"/> 60 days
---	--	--	---

Recommendation: Purchase a Compute Savings Plan at a commitment of \$2.40/hour

You could save an estimated \$1,173 monthly by purchasing the recommended Compute Savings Plan.

Based on your past **60 days** of usage, we recommend purchasing a Savings Plan with a commitment of **\$2.40/hour** for a **3-year term**. With this commitment, we project that you could save an average of **\$1.61/hour** - representing a **40%** savings compared to On-Demand. To account for variable usage patterns, this recommendation maximizes your savings by leaving an average **\$0.04/hour** of On-Demand spend.

Before recommended purchase	After recommended purchase (based on your past 60 days of usage)	
Monthly On-Demand spend ⓘ \$2,955 (\$4.05/hour) <small>Based on your On-Demand spend over the past 60 days</small>	Estimated monthly spend ⓘ \$1,782 (\$2.44/hour) <small>Your recommended \$2.40/hour Savings Plans commitment + an average \$0.04/hour of On-Demand spend</small>	Estimated monthly savings ⓘ \$1,173 (\$1.61/hour) <small>40% monthly savings over On-Demand \$2,955 - \$1,782 = \$1,173</small>

This recommendation examines your usage over the past 60 days (including your existing Savings Plans and EC2 Reserved Instances) and calculates what your costs would have been had you purchased the recommended Savings Plans. See applicable rates for Savings Plans [here](#). To generate this recommendation, AWS simulates your bill for different commitment amounts and recommends the commitment amount that provides the greatest estimated savings. [Learn more](#)

Recommended Compute Savings Plans

[Download CSV](#) [Add selected Savings Plan\(s\) to cart](#)

x	Term	Payment option	Recommended commitment	Estimated hourly savings
<input checked="" type="checkbox"/>	3-year	All upfront	\$2.40/hour	\$1.61 (40%)

*Average hourly spend and minimum hourly spend based on your current on-demand spend for the given instance family.

Multiple Savings Plans can be purchased so don't feel obligated to make one large commitment to support all your workloads.

- Savings Plans are a simple way to reduce your spend, but remember that unlike Reserved Instances, they can't be sold, nor is there any sort of volume discount opportunity. Consider centralizing the cost commitment function across your organization, and provide advice and guidance to your application owners or business units to make sure they select the cost saving strategy that most benefits the Organization as a whole.
- Perform regular cost reviews and pay attention to long-term trends in commitment utilization and coverage. While some individual or ephemeral workloads such as development environments or sandboxes may not make sense for Service Plans or Reserved Instances, it may be worth committing to a percentage of these workloads in aggregate.

For example, the hourly spend across all development accounts may vary from \$50 to \$100 an hour. If that spend rate is consistent for many months, consider a savings plan commitment of 50-80% of minimum sustained usage, or \$25 to \$40 an hour.

- Tag your Reserved Instance and Savings Plan purchases. Tie each back to a specific business justification and make sure it still applies when it comes time to consider renewal.

Launching Control Tower in the Real World

For the past year, Logicworks has launched nearly a dozen multi-account solutions for a wide variety of customers. Control Tower is a perfect toolset for any company that needs to segregate business units or SaaS tenants while maintaining central billing and security baselines. And companies that we've worked with have been pleased at the result: a secure, well-organized account structure that can expand with their company.

Here are just a few of the multi-account projects that we've worked on in the last 12 months:

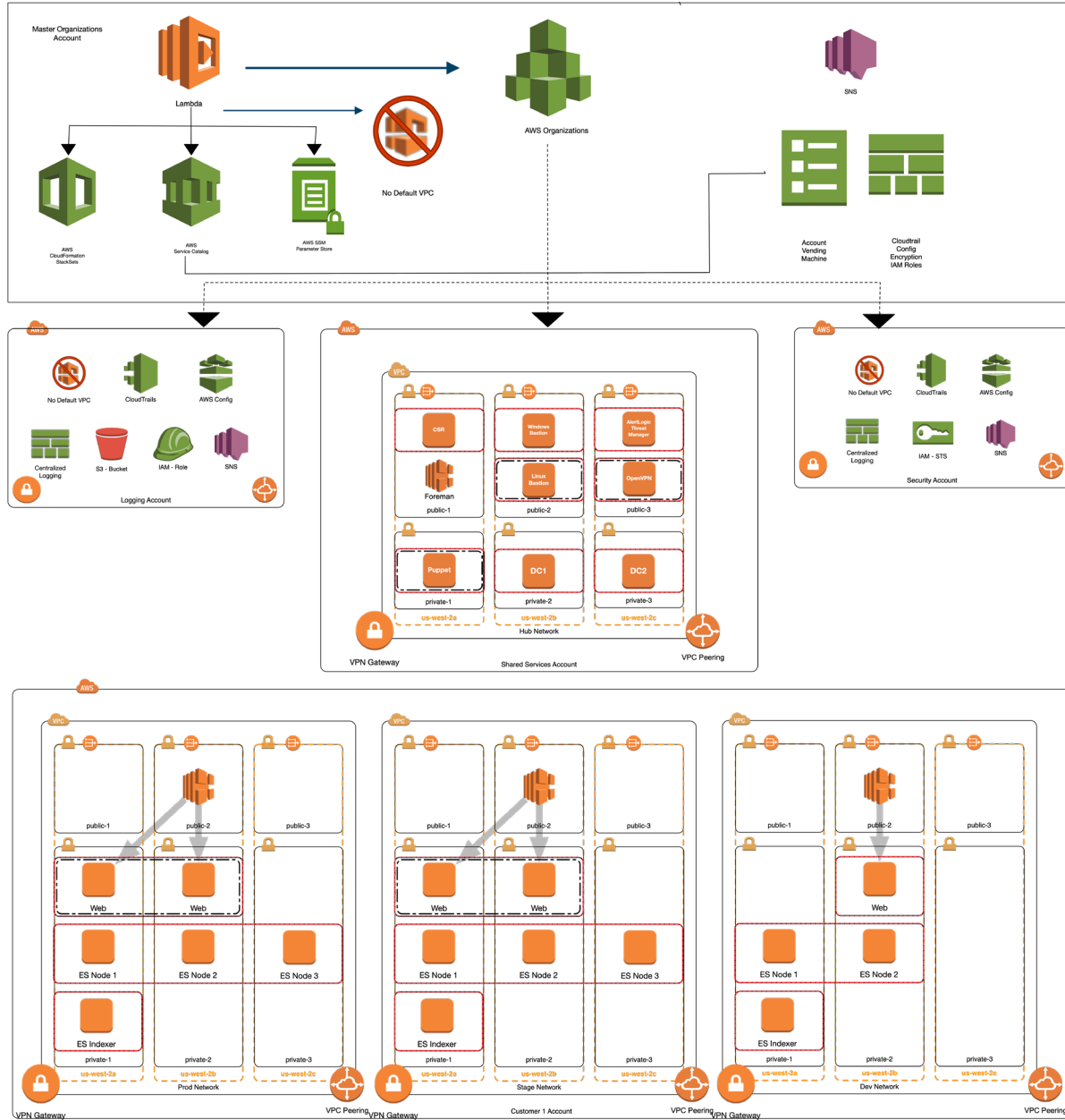
- A financial investment management firm used a landing zone to provide dedicated accounts for each portfolio manager or team, with automated baselines and security tooling by default.

- A franchise-based service delivery organization used a landing zone to provide accounts for every business unit and application development lifecycle
- A billing management SaaS provider used a landing zone to deploy dedicated accounts per end tenant for clear infrastructure segmentation by customer. This also allows them to maintain different security and compliance requirements for each customer, if necessary.

Control Tower is a perfect toolset for any company that needs to segregate business units or SaaS tenants while maintaining central billing and security baselines.

Sample AWS Control Tower Architecture

The following are actual architecture diagrams from a project Logicworks recently completed with a SaaS company. Each account has its own diagram, but for the purposes of this guide, we've provided the overall account structure and a look at network flow between various critical components.



Network Flow Detail

This architecture diagram shows how information flows from the on-premises datacenter through the Network Accounts to the App Account.

Data flow from on-premises to AWS:

- Data flows back and forth from their on-premises datacenter through an AWS Direct Connect (dedicated network connection) to the Network Account
- The Direct Connect Gateway can transfer connections to multiple VPCs, like a Transit Gateway
- On the App account, a Virtual Private Interface acts as the endpoint from the Direct Connect

End user access to the applications:

- From the internet and Akamai (a CDN) to the load balancer, which distributes the traffic to the instances that are contained in private subnets. This represents all inbound traffic from external users to the eCommerce website assets (all static content)

Logs flow from every account (including the Network Account, Sandbox, Dev/Test, QA/STG, and Production accounts) to the Log Account using a Lambda function and IAM access

Cross-account IAM permissions are allowing access from services in the Shared Services account to the App Account

How to Deploy This Architecture

Since AWS Control Tower is a multi-account solution, it's not possible to give you a CloudFormation template, as we will for other architectures in this Guide. Control Tower isn't really an AWS service in its truest form. It has no API and you can't create it with CloudFormation. It's just a wrapper for other AWS services through the console.

To launch a Control Tower, navigate in the AWS console to <https://console.aws.amazon.com/controltower>. Once there, you can pick your desired home region, provide details about core OUs, review service permissions, and launch Control Tower.

We recommend checking out the extensive AWS documentation [here](#), in order to launch a Control Tower or reach out to an approved AWS partner like Logicworks.

Summary

In this guide, we discussed the basics of AWS Control Tower and outlined a few best practices. As an implementation example, we introduced the AWS Control Tower solutions that we used to help customers deploy real-life applications.

A multi-account architecture is an ideal solution if you're migrating a large, complex set of applications to AWS. AWS Control Tower is

meant to help reduce the complexity of building and managing a multi-account structure long-term.

Need help architecting a custom solution or managing your AWS Control Tower? Our team of AWS experts have designed hundreds of custom AWS environments and can help you get the most out of AWS. [Contact Logicworks](#) to learn more.

About Logicworks

Logicworks is a leading provider of AWS migration and managed services. As an AWS Premier Consulting Partner, we have helped hundreds of companies architect, migrate, and manage custom AWS environments. We specialize in complex, highly regulated workloads for healthcare, finance, and retail and have earned HIPAA, HITRUST, PCI-DSS, SOC1, SOC2, and ISO 27001 certification. Learn more about Logicworks at www.logicworks.com.