aws marketplace

# Implementation Guide:
# Integrating Snyk Container with AWS Control Tower

# Table of Contents

# Foreword

Snyk is an AWS Advanced Technology Partner with DevOps & Security Competencies, and Amazon Linux 2 & Lambda Service Ready Validations. Snyk is a contributor to AWS Modernization Workshops and natively integrates with AWS CodePipeline allowing customers to build automated security controls into their deployment pipeline without having to leave the AWS console, bringing the Snyk experience directly to AWS users, and empowering them to more efficiently find and fix vulnerabilities in open source code when building cloud-native applications on AWS.

Snyk's developer security solutions enable modern applications to be built securely, empowering developers to own and build security for the whole application, from code & open source to containers & cloud infrastructure. Snyk Container is a security solution that provides an Amazon Elastic Container Registry (ECR) on the AWS cloud. Implementing this solution, you can monitor your container images hosted on ECR with Snyk.

The purpose of this AWS Implementation Guide is to enable customers to seamlessly activate, deploy and configure the Snyk Container integration to Amazon ECR with multi-account support for environments managed by AWS Control Tower while taking full advantage of the resources pre-configured by AWS Control Tower as part of the initialization.

## Solution overview and features

Snyk Container is a security solution designed to help developers find and fix vulnerabilities in cloud native applications. Snyk's seamless integration into the developer workflow, with continuous monitoring of applications in production, empowers developers to continue to release fast, while ensuring secure code.

The Snyk Container for AWS Control Tower solution is for developers, DevOps, DevSecOps, security teams, and other roles within an organization that build, deploy, and maintain serverless applications or container images that use Amazon Elastic Container Registry (Amazon ECR).

With Snyk Container, you can:

- ✔ Automate base image upgrades to quickly resolve numerous vulnerabilities
- ✔ Continuous monitoring for new vulnerabilities
- ✔ Prioritize fixes based on context and exploitability
- ✔ Discover issues in open-source libraries
- ✔ Match vulnerabilities to Dockerfile commands

# Architecture diagram

Snyk Container integrates with Amazon ECR to help you import projects and monitor containers for vulnerabilities. Snyk Container tests your imported projects for known security vulnerabilities at a frequency that you control. Integration with Amazon ECR is available for all pricing plans. For more information, see Snyk Container.

This Snyk Container integration makes use of AWS CloudFormation to provision an Amazon ECR container registry through AWS Control Tower. When the CloudFormation template is deployed to a region a single-use AWS Lambda function is deployed which connects to Snyk to establish the initial integration. A cross-account IAM role is used thereafter to handle access between Snyk and the newly deployed Amazon ECR.

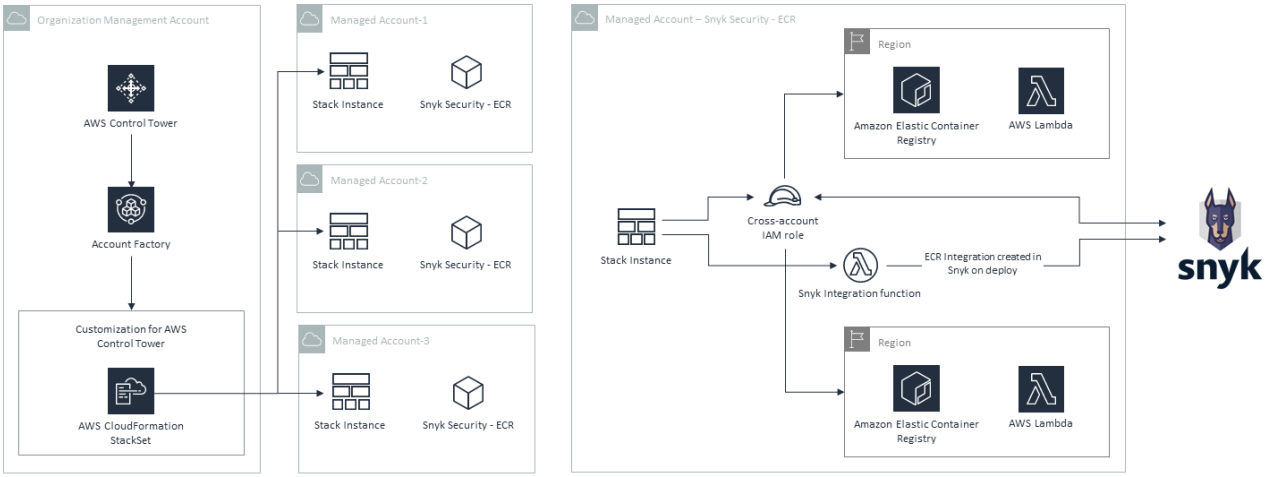



*Figure 1 Snyk Container for AWS Control Tower Architecture Diagram*

To deploy the Snyk Container integration solution in AWS Control Tower, the Customizations for AWS Control Tower solution is used. The Customizations for AWS Control Tower solution combines AWS Control Tower and other highly-available, trusted AWS services to help customers more quickly set up a secure, multi-account AWS environment using AWS best practices.

You can easily add customizations to your AWS Control Tower landing zone using an AWS CloudFormation template and service control policies (SCPs). You can deploy the custom template and policies to individual accounts and organizational units (OUs) within your organization. For this solution, we will deploy AWS CloudFormation templates only. This solution integrates with AWS Control Tower lifecycle events to ensure that resource deployments stay in sync with your landing zone. For example, when a new account is created using the AWS Control Tower account factory, the solution ensures that all resources attached to the account's OUs will be automatically deployed.

## Pre-requisites

The Snyk Container for AWS Control Tower solution requires either a Business or Enterprise Snyk account. You may purchase Snyk Business or Snyk Enterprise Plans through the AWS Marketplace as described in the *Deployment and Configuration Steps* section below. Additionally, the authentication token used for deploying the Snyk Container for AWS Control Tower solution must have Group access within Snyk. An individual user authentication token with sufficient access may be used, but it is preferable to use a group level Service Account token with *admin* access. Refer to Snyk's Group documentation for more information.

Before launching Snyk Container for AWS Control Tower solution, you must sign in to the AWS Management console with IAM permissions for the resources that the CloudFormation templates deploy. The *AdministratorAccess* managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions. For more information, see AWS managed policies for job functions.

If necessary, request service quota increases for the following resources. You might need to request increases if your existing deployment currently uses these resources and if this Quick Start deployment could result in exceeding the default quotas. The Service Quotas console displays your usage and quotas for some aspects of some services. For more information, see What is Service Quotas? and AWS service quotas.

Deploy the Customizations for AWS Control Tower solution before you proceed further. While launching the solution, change the value of ***AWS CodePipeline Source*** to *AWS CodeCommit* as the instructions in this guide are based on AWS CodeCommit. The default value for Amazon S3.

Before you implement this solution, we recommend that you become familiar with AWS CloudFormation, IAM, Amazon Elastic Container Registry, and AWS Lambda.

If you are new to AWS, see Getting Started with AWS

For additional information on AWS Marketplace

To get started with AWS Control Tower, check out the Control Tower User Guide

# Deployment and Configuration Steps

**Step 1.1: Subscribe to Snyk: Developer Security Platform (Business and Enterprise Tiers) on AWS Marketplace.**

Locate the Snyk listing in the AWS Marketplace.

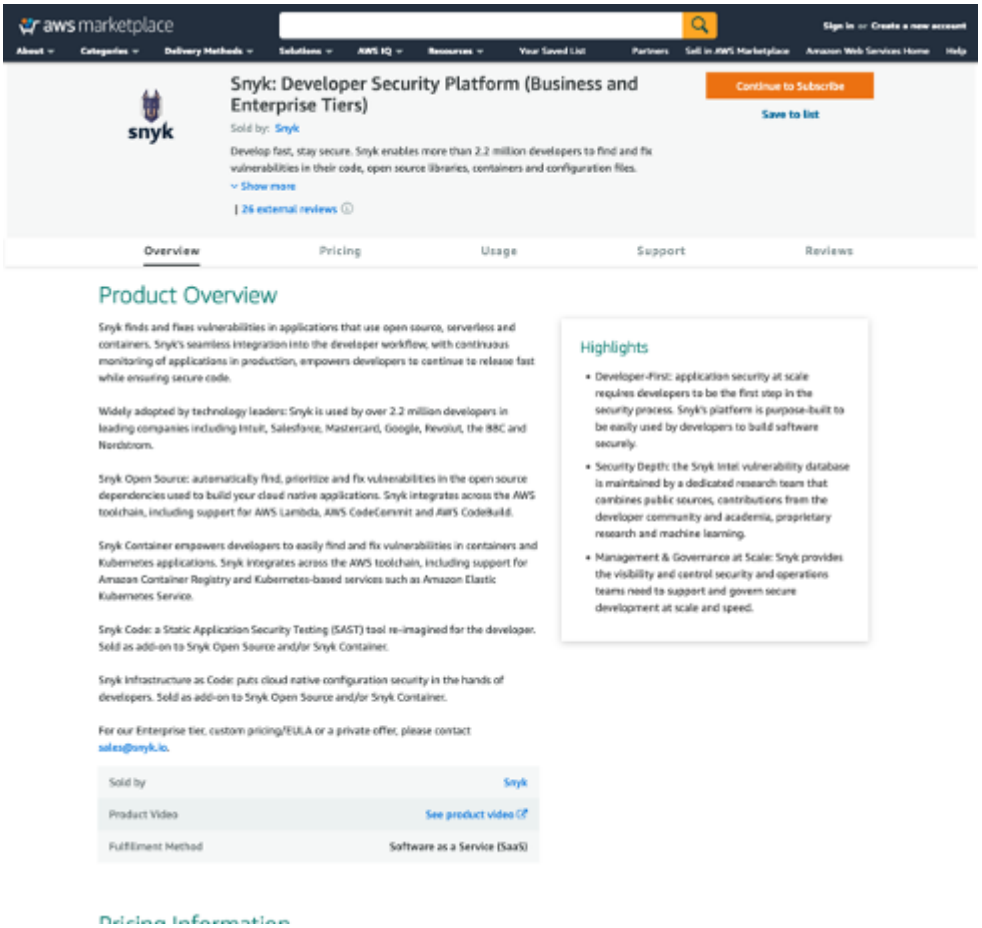
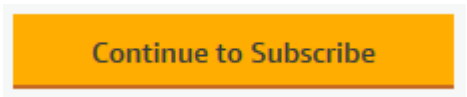



*Figure 2 Snyk: Developer Security Platform in the AWS Marketplace*

Click on the **Continue to Subscribe** button.



### Step 1.2: Configure Contract Duration and Renewal

In the new screen, you can configure your contract. You can select the **Contract Duration** and set the **Renewal Settings**.

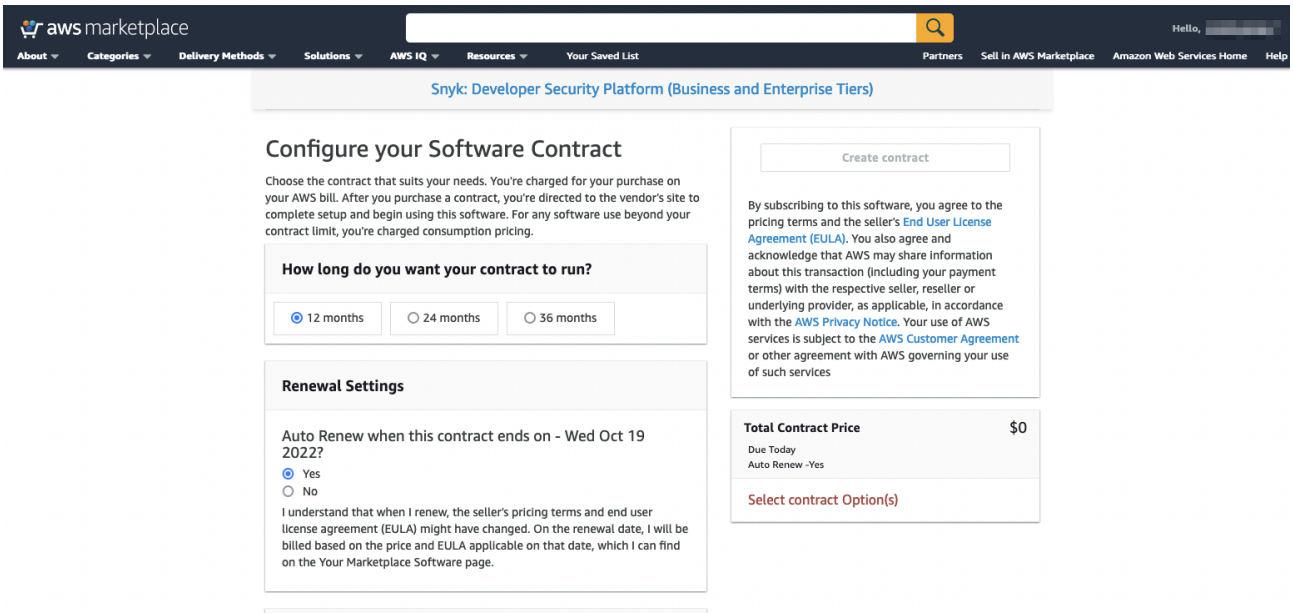



*Figure 3 Snyk contract configuration in the AWS Marketplace*

### Step 1.3: Select Contract Options

Select the appropriate Contract option based on your needs. The solution discussed in this document works on any active contracts listed in this screen.

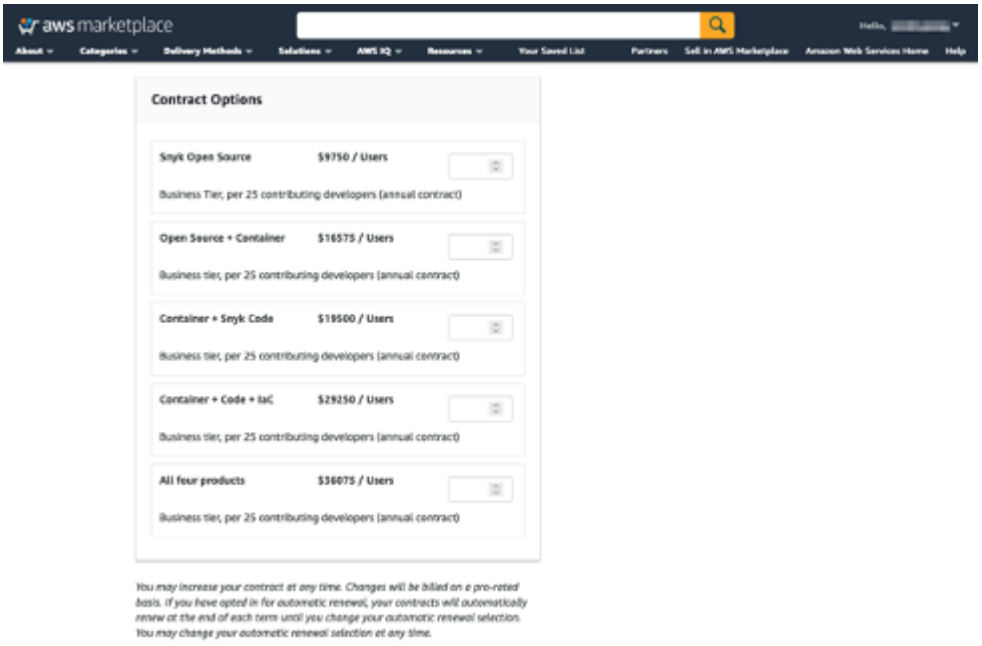



*Figure 4 Snyk contract options in the AWS Marketplace*

**Step 1.4: Create the Contract and Pay**

Once you have configured your contract, you can click on the Create contract button.

You will be prompted to confirm the contract. If you agree to the pricing, select the **Pay Now** button.

**Step 1.5: Set up Account**

Log in to your Snyk Account, and obtain either your API Token, as shown in Figure 5, or create a Service Account as shown in Figure 6.

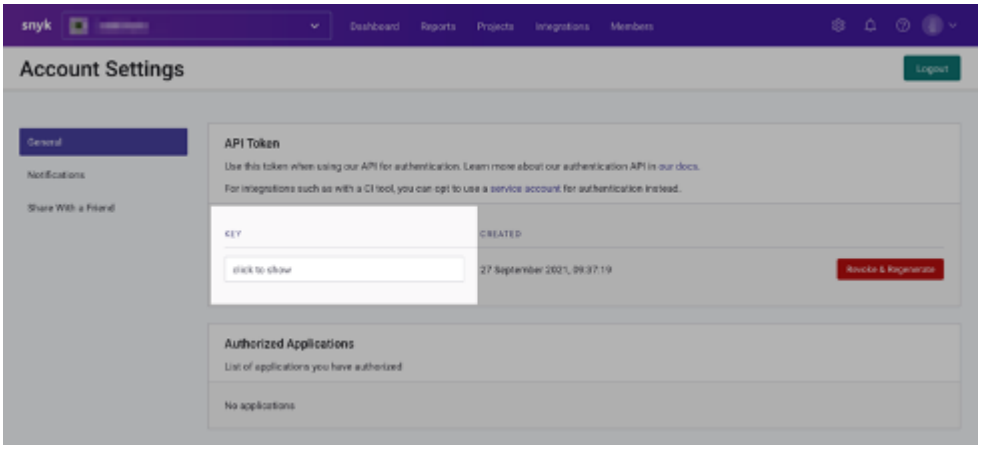



*Figure 5 Snyk Account settings page*

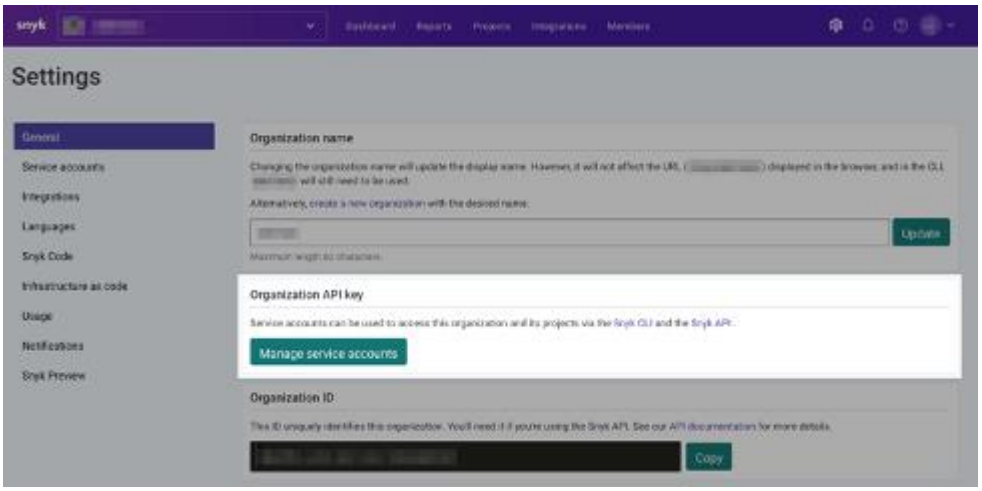



*Figure 6 Snyk Organization settings page*

**Snyk Account Access**

The Snyk AWS Account is required to assume a role in your account to function. Please be aware that this deployment grants Snyk the ability to assume an IAM role in your account. To continue, please use **198361731867** as the account ID for the Snyk AWS Account ID parameter when prompted.

## Deploying Snyk Container for AWS Control Tower

To deploy Snyk in AWS Control Tower environments, we leverage Customizations for AWS Control Tower solution. The Customizations for AWS Control Tower solution combines AWS Control Tower and other highly-available, trusted AWS services to help customers more quickly set up a secure, multi-account AWS environment using AWS best practices.

You can easily add customizations to your AWS Control Tower landing zone using an AWS CloudFormation template and service control policies (SCPs). You can deploy the custom template and policies to individual accounts and organizational units (OUs) within your organization. This solution integrates with AWS Control Tower lifecycle events to ensure that resource deployments stay in sync with your landing zone. For example, when a new account is created using the AWS Control Tower account factory, the solution ensures that all resources attached to the account's OUs will be automatically deployed.

This solution is widely deployed by AWS Control Tower customers who require additional customizations in their environments. Customizations include, creating additional detective or prevention guardrails, and enabling additional services. As mentioned in the pre-requisites, this customizations solution supports both Amazon S3 and AWS CodeCommit repository as a source for the customization templates and the configuration file (***manifest.yaml***). For this solution, we use AWS CodeCommit repository. For additional implementation details and options, refer to Customizations for AWS Control Tower Implementation Guide.

You need to update the configuration file using the sample provided in this implementation guide. Additional details in the instructions below.

**Step 2.1: Launch Customizations for AWS Control Tower solution in your Management Account**

- Login to your AWS Control Tower Management account.
- If you already launched Customizations for AWS Control Tower solution and using for existing customizations, skip next step in this section and edit your existing repository.
- Launch the Customizations for AWS Control Tower solution using this link.
- In *Create Stack page*, choose **Next**.

- In *Specify stack details* page,
    - Enter a **Stack name** you prefer.
    - Change **AWS CodePipeline Source** to *AWS CodeCommit*. The default value is Amazon S3. If you choose to use Amazon S3 as configuration source, please refer to Using Amazon S3 as the Configuration source documentation for usage details.
    - Leave the remaining default values and choose **Next.**
- In *Configure stack options* page, choose **Next.**
- In *Review* page, select check box **I acknowledge that AWS CloudFormation might create IAM resources with custom names** and choose **Create stack.**

Wait for the stack status change to CREATE_COMPLETE. For additional information related to the solution, refer to Customizations for AWS Control Tower solution page.

**Step 2.2: Update the configuration file with Snyk integration**

- Navigate to AWS CodeCommit Console, and choose **custom-control-tower-configuration**.
- Use your favorite method to Edit the file *manifest.yaml* and add below content and commit the changes. Read the comments in the content below and make appropriate changes depending on your environment.

```
---
region: us-east-1 # Control Tower Home Region
version: 2021-03-15

resources:
  # Roles needed only for Snyk security solution testing

  - name: snyk-ecr-integration
    description: 'Snyk quickstart deployment'
    resource_file: https://aws-quickstart.s3.us-east-1.amazonaws.com/quickstart-snyk-security/templates/snyk-ecr-auto.template.yaml # update if you copied the content to local S3 bucket.
    deploy_method: stack_set
    #Apply to the following OU(s)
    deployment_targets:
      organizational_units:
        # UPDATE THIS SECTION: with OU names you have in your environment.
        # All the AWS Accounts existing/future will have the Snyk solution deployed.
        - Security
        - Sandbox
    parameters:
      - parameter_key: SnykGroupId
        parameter_value: 'aaaabbbb-cccc-dddd-eeee-ffffgggghhhh' # Get your unique value from Snyk portal
      - parameter_key: SnykAuthToken
        parameter_value: 'iiiijjjj-cccc-dddd-eeee-ffffgggghhhh' # Get your unique value from Snyk portal.
      - parameter_key: SnykOrgPatternPrefix
        parameter_value: 'yourorgname' # Update with name to use as a prefix in Snyk portal
      - parameter_key: SnykAWSAccountNumber
        parameter_value: '198361731867' # Do not change this unless suggested by Snyk team.
      - parameter_key: ECRResourceARN
        parameter_value: '*'
      - parameter_key: QSS3BucketName
        parameter_value: 'aws-quickstart' # Update if you copied the content to local S3 bucket.
      - parameter_key: QSS3KeyPrefix
```

```
        parameter_value: 'quickstart-snyk-security/' # Update if you copied the content to
local S3 bucket.
      - parameter_key: QSS3BucketRegion
        parameter_value: 'us-east-1' # Update the region if you copied the binaries to
local S3 bucket.
    regions:
      - us-east-1 # This solution needs to be deployed only in any one region. Update as
needed.
```

- This commit will trigger the AWS CodePipeline. Check the pipeline status from AWS CodePipeline Console
- Wait for all 4 stages to complete with status **Succeeded**.

You successfully deployed the Snyk integration solution in your AWS Control Tower environment. Depending on the configuration you mentioned above, all new accounts provisioned will now be automatically integrated with Snyk.

**Step 2.3: Verify the solution deployment**

- Navigate to AWS CloudFormation Stacksets console and find Stackset with name *CustomControlTower-<resource-name>.* According the example provided in this document, it is *CustomControlTower*-snyk-ecr-integration.
- Choose **Stack instances** to see all the AWS accounts to which this solution is deployed.

## What to Expect

**Step 3.1: Changes to your Snyk account**

The Snyk Security – ECR solution creates a new Organization within your provided Snyk group with which to connect the ECR integration.

By default, the newly created Organization will have a semi-random name based on the *SnykOrgPatternPrefix* parameter. It is possible to modify the Organization's name, to create a more descriptive one, from within Snyk's interface. To do so:

1. Log in to your Snyk account.
2. Click the gear icon (  ) near the top right corner of the Dashboard.
3. Replace the text displayed in the field labelled **Organization Name**
4. Click **Update**

**Step 3.2: How to use**

Deploying the Snyk Security – ECR solution automatically creates the integration for your Snyk Organization. Once deployed, you can add repositories to scan with Snyk by performing the following steps:

1. Log in to your Snyk account.
2. Go to **Projects**, select **Add projects**, and then select **Amazon ECR**.
3. Select single or multiple images.
4. Select **Add selected repositories.**

Congratulations, you have successfully deployed Snyk Security for AWS Control Tower solution to automatically register the AWS accounts that you create using Account Factory. You can further add the ECRs in the accounts in the Snyk portal and automatically scan for the vulnerabilities.

## Solution Estimated Pricing

This solution can be deployed at no additional cost for customers participating in a Snyk Business or Snyk Enterprise Plan. To learn more about Snyk Pricing & Plans please our pricing page for more details.

## FAQs

Support for new customers including onboarding sessions, office hours, and live hacking are available through our Snyk Community.

## Additional resources

### AWS resources

- Getting Started Resource Center
- AWS General Reference
- AWS Glossary

### AWS services

- AWS CloudFormation
- IAM
- Amazon Elastic Container Registry (ECR)
- AWS Lambda

Snyk

- Snyk User Documentation
- Snyk Support Portal
- Snyk Learn

## Partner contact information

For any questions about this and other solutions, please contact us here.