PRISMA CLOUD AppSec Camp STUDENT GUIDE

Lab Introduction

Thank you for joining today's AppSec camp. The lab will focus on application security throughout the software development lifecycle. Before we begin the lab let's start with a brief overview of the lab scenario to help frame the context.

Scenario

The Exampli Corp product group is rushing to finish a banking app for their customer Bank of Anthos in time for the holiday season when spending patterns spike. Up against unrealistic deadlines the development and infrastructure teams are working around the clock to get their app built, tested, and released to hit their deadlines.

Fortunately, Exampli Corp recently integrated Prisma Cloud into their development lifecycle adopting shift left security from code to cloud. Once in production Exampli Corps operations and security teams continue to leverage Prisma Cloud to monitor and protect runtime resources, reduce the attack surface, and enforce least privilege.

Will The Exampli Corp team take the time to build a secure app? Or will the stress of completing the Bank of Anthos app in time for the holiday season lead to mistakes?

Resources

Vulnerable Repo :

In this lab we leverage some intentionally vulnerable code repository. To learn more about the project and its contributors visit the link below :

- https://github.com/bridgecrewio/supplygoat

Bank of Anthos Application :

In this lab there is a mock banking application that is used as the application the Exampli Corp is building. To learn more about the project and its contributors visit the link below :

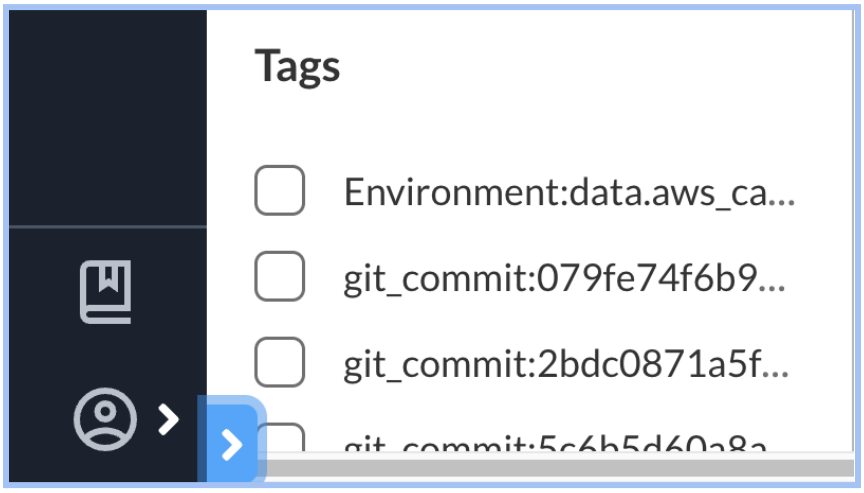
- https://github.com/GoogleCloudPlatform/bank-of-anthos

Exercise

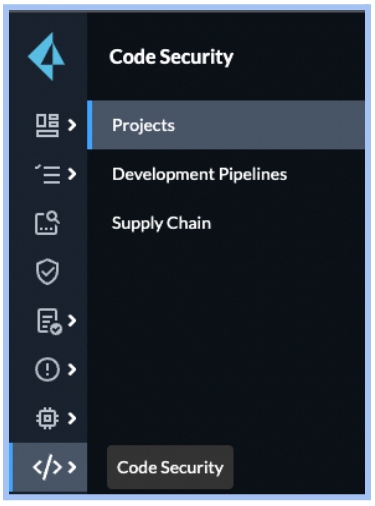
Let's begin by exploring the power of shifting security left with Infrastructure as code scanning. As infrastructure is being defined as code security must be integrated with the tools developers use. In this exercise, we will take a look at some IaC templates within Exampli Corp's repository.

Software Composition Analysis:

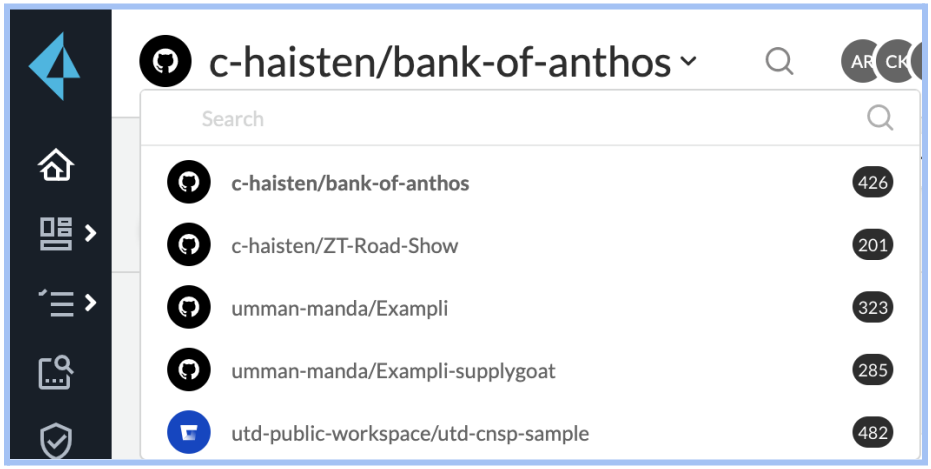
1. Login to Prisma Cloud.
2. Use the credentials provided by your Instructor to authenticate.
3. Use the navigation pane on the left hand and click the **blue arrow** on the lower left side of the UI to open up the navigation pane and move between the different modules within Prisma Cloud.



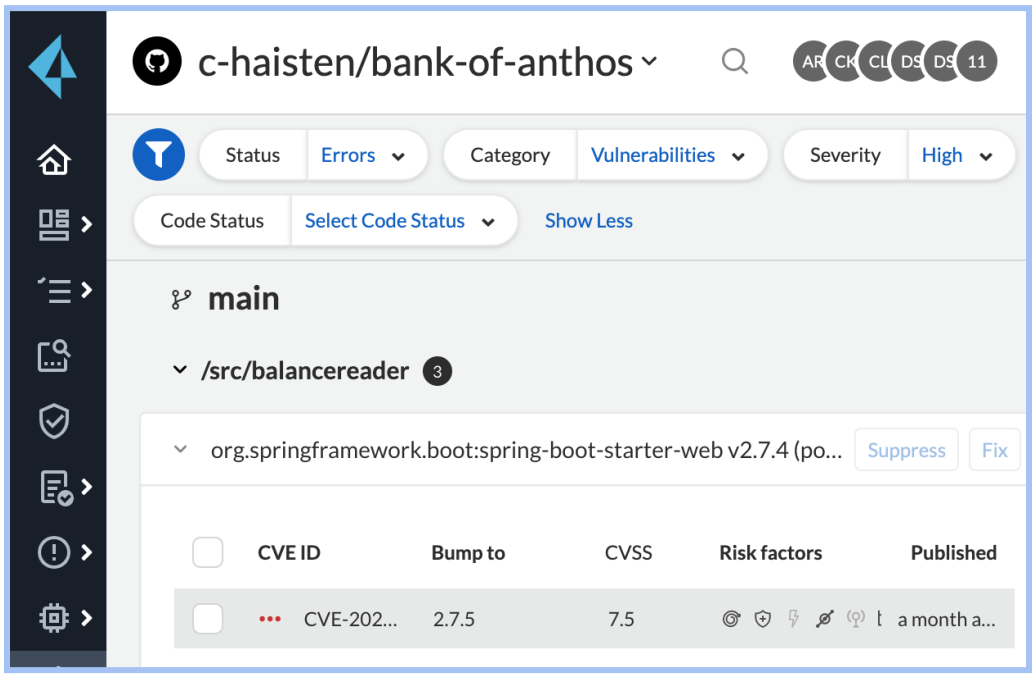
4. Next, use the navigation pane to select the **Code Security** module and then select **Projects.**



5. Ensure that you select the **C-haisten/bank-of-anthos** repository.



6. Once you have selected the correct repository, it will show you all the security incidents found in the code of that branch. Let's check to see if there are any serious vulnerabilities by filtering **Category -> Vulnerabilities** and **Severity -> High.**

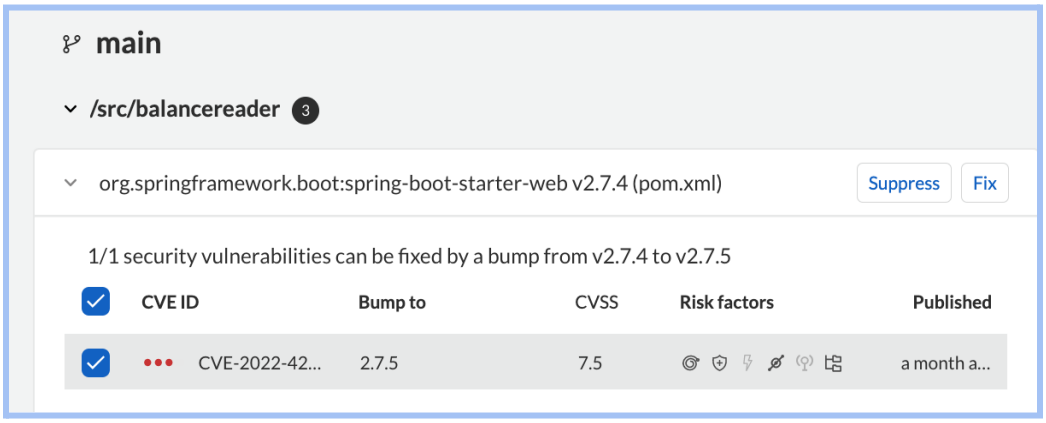


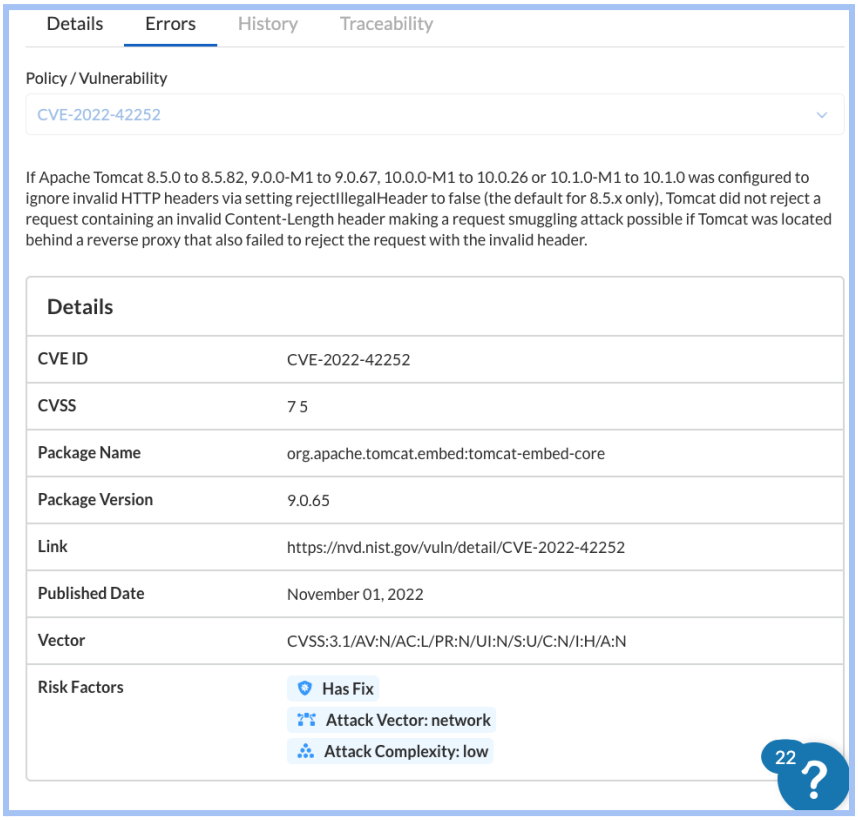
7. Next, we will select a CVE ID to investigate the vulnerability and determine what next steps to take. Prisma Cloud gives us several options for how to interact with the vulnerability.

   Administrators can click suppress, or fix. The suppress button allows you to dismiss all incidents that fall under that specific policy violation. The fix button will allow you to remediate the vulnerability via a bump fix. Here we can see that the vulnerability can be fixed easily by bumping from v2.7.4 to 2.7.5.

   Suppress and Fix require increased RBAC that is not available in this lab

8. You can also click the “Details” or “Errors” tab on the right side of the page to get some more information on the dangers of the specific vulnerability.

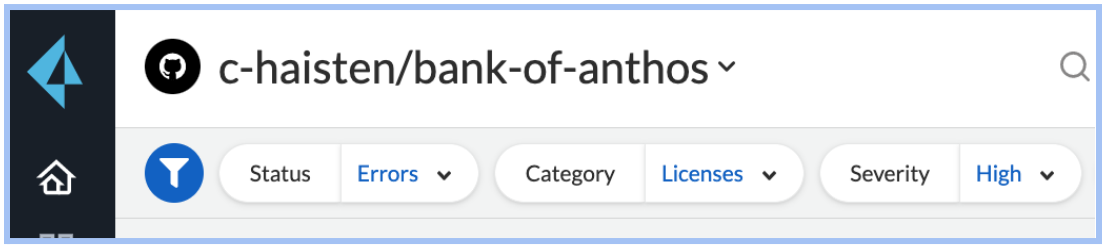




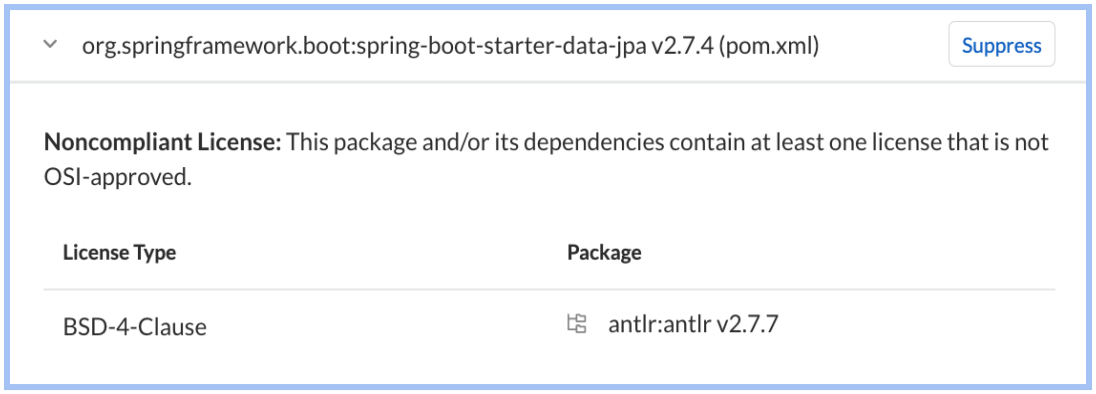
9. This CVE has a CVSS score of 7.5! It makes Exampli Corp’s Bank of Anthos app vulnerable to a request smuggling attack.

   On this page Exampli Corp developers can gain context on the CVE and click on the link to NIST providing them with all the details and documentation regarding the vulnerability.

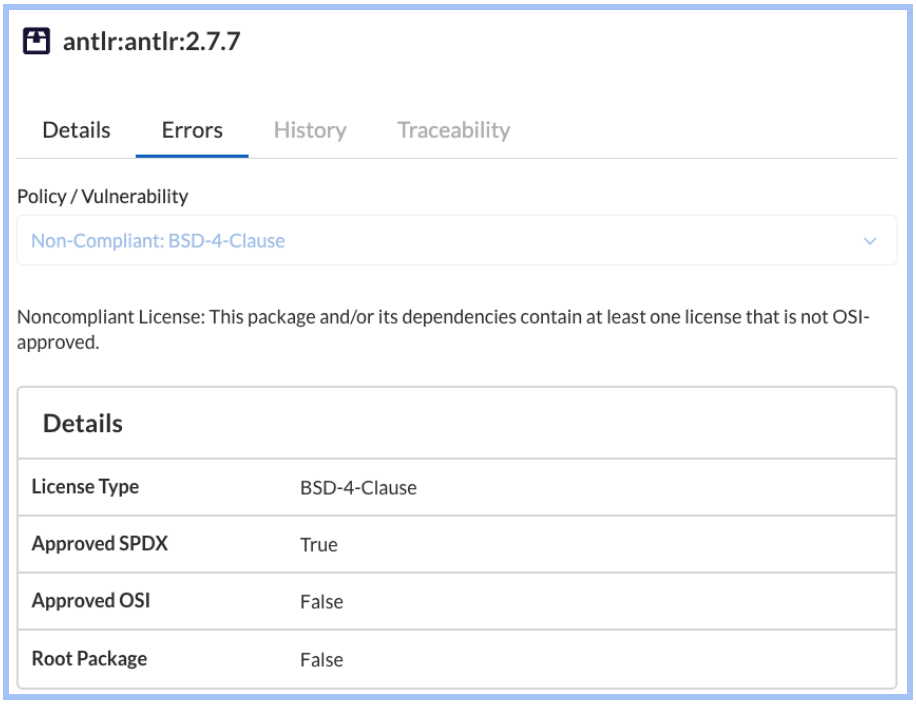
10. In Prisma Cloud licenses are scanned in parallel to vulnerability scanning. This means that Exampli Corp developers can make sure they are remaining compliant when working with open source packages. Adjust your filter to **Category -> Licenses** and let's see if there are any licensing issues in the Bank of Anthos Repo.



11. Now, let's scroll down and look at some of the licensing issues identified by Prisma Cloud.



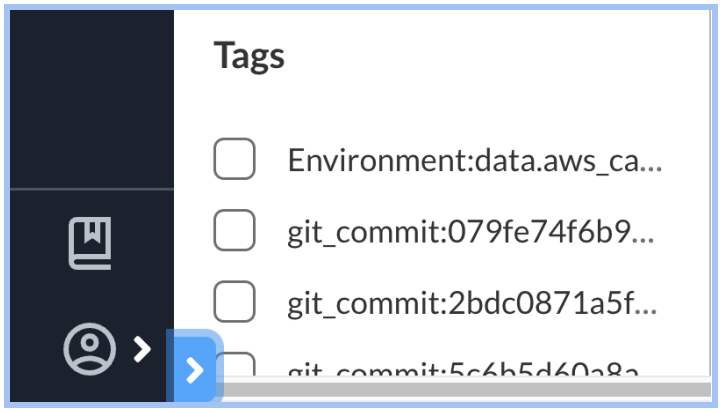
12. By Clicking on the license type Prisma Cloud provides developers with critical information and identifies each policy violation as a single error.



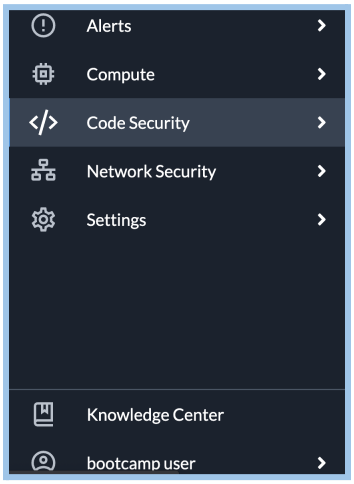
Now that we have found some vulnerabilities and licensing violations in our Application code, let's take a look at how Prisma Cloud can give us visibility to the package manager files that comprise applications by taking a look at the supply chain and software composition analysis (SCA).

Supply Chain Security

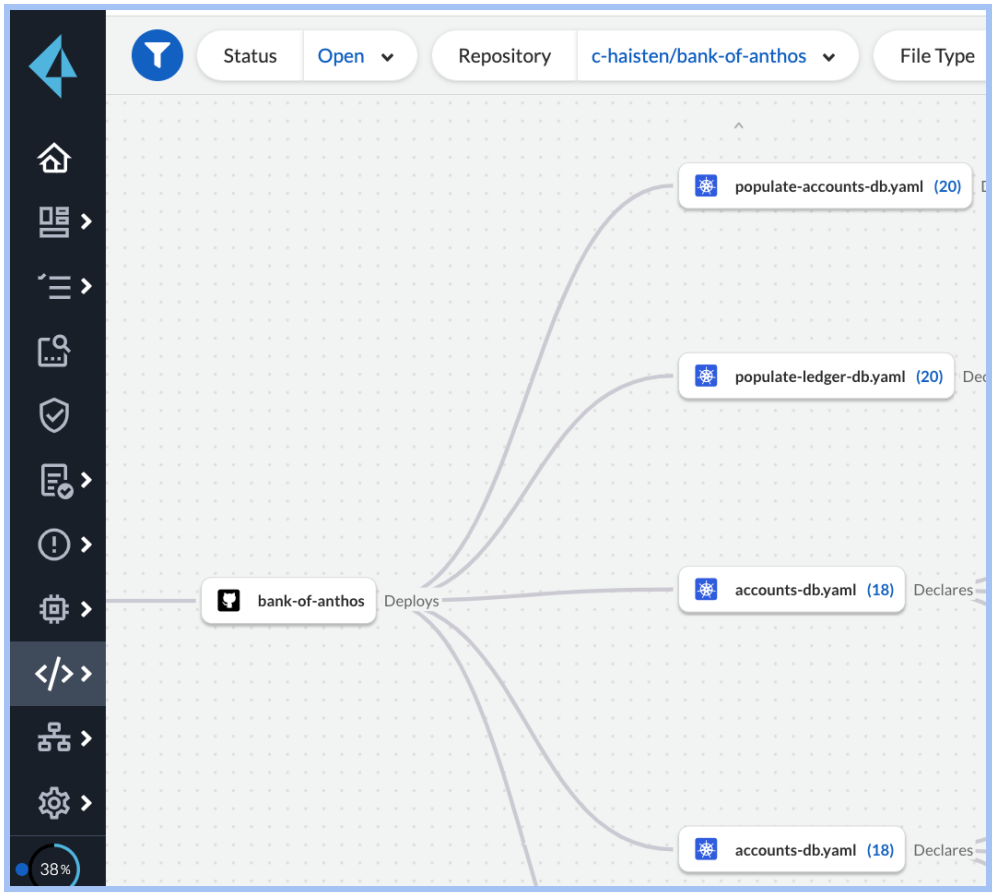
1. Use the navigation pane on the left hand and click the **blue arrow** on the lower left side of the UI to open up the navigation pane and move between the different modules within Prisma Cloud.



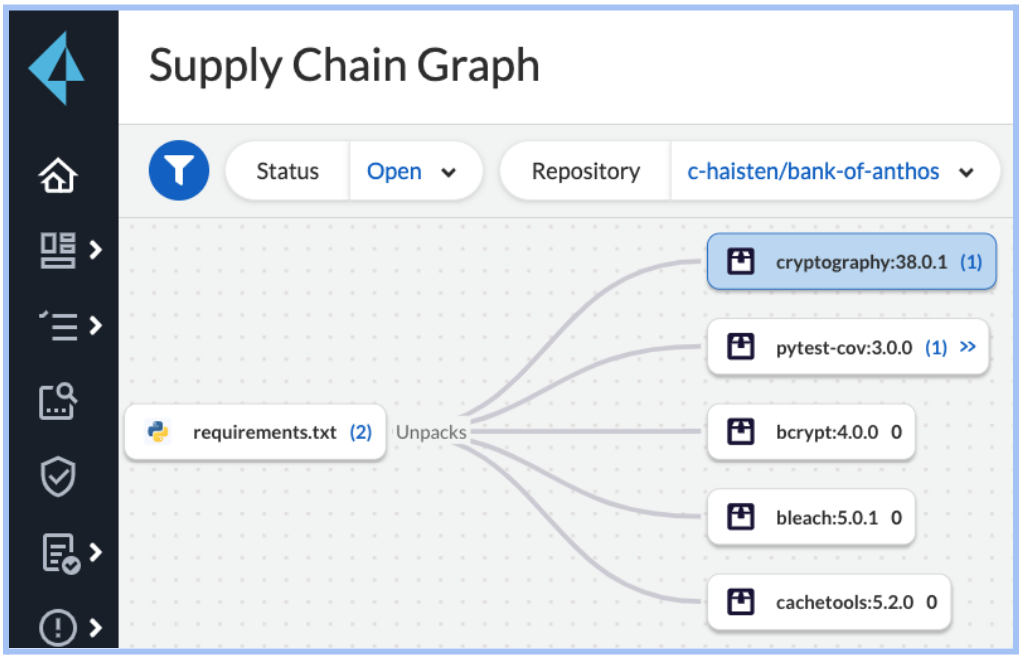
2. Next, use the navigation pane to select the **Code Security** module and select **Supply Chain**.



3. Use the Supply Chain Graph to view the relationships between different IaC templates and the types of infrastructure and services they provision. Be sure to select the bank-of-anthos repository from the drop-down window in the filters at the top.



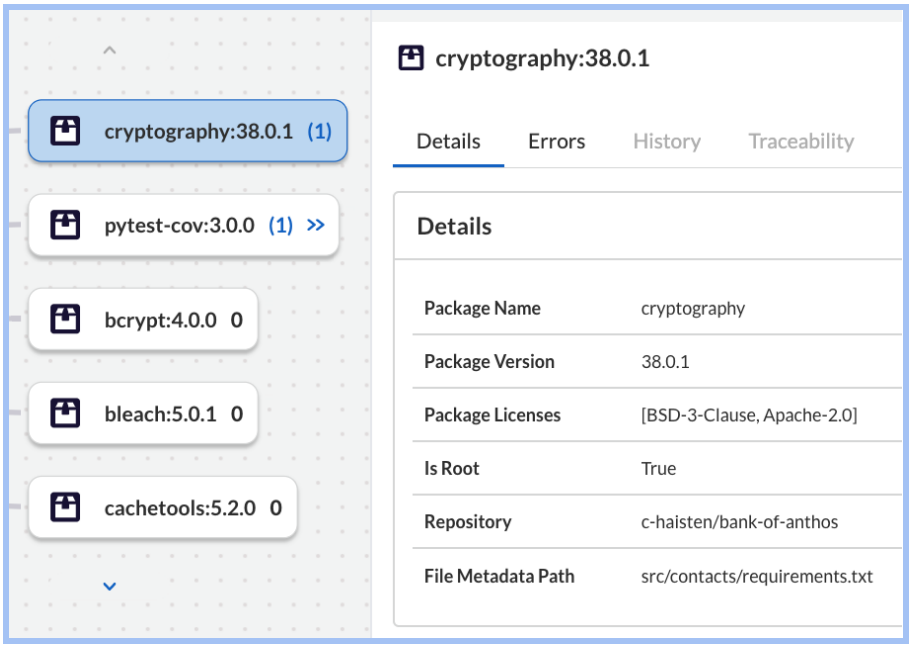
4. Next, take a deeper look at the **requirements.txt** file. Feel free to test the filters on the left side of the UI and search bar at the top to quickly locate templates.



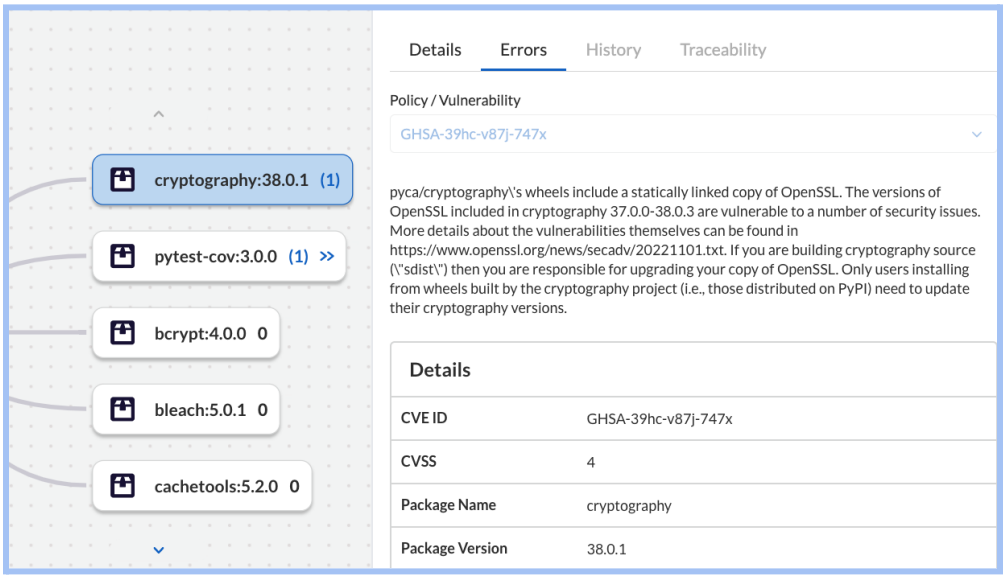
5. Once you have found the **requirements.txt** file click on the first package that is unpacked, **cryptography:38.0.1**. Notice on the right side of the UI there is additional information about this resource.

   Here we can quickly identify valuable information like the package version, license type and privileges.

6. Your screen should look similar to the screenshot below.



7. Next, click on the **Errors** tab to investigate policy violations and vulnerabilities.



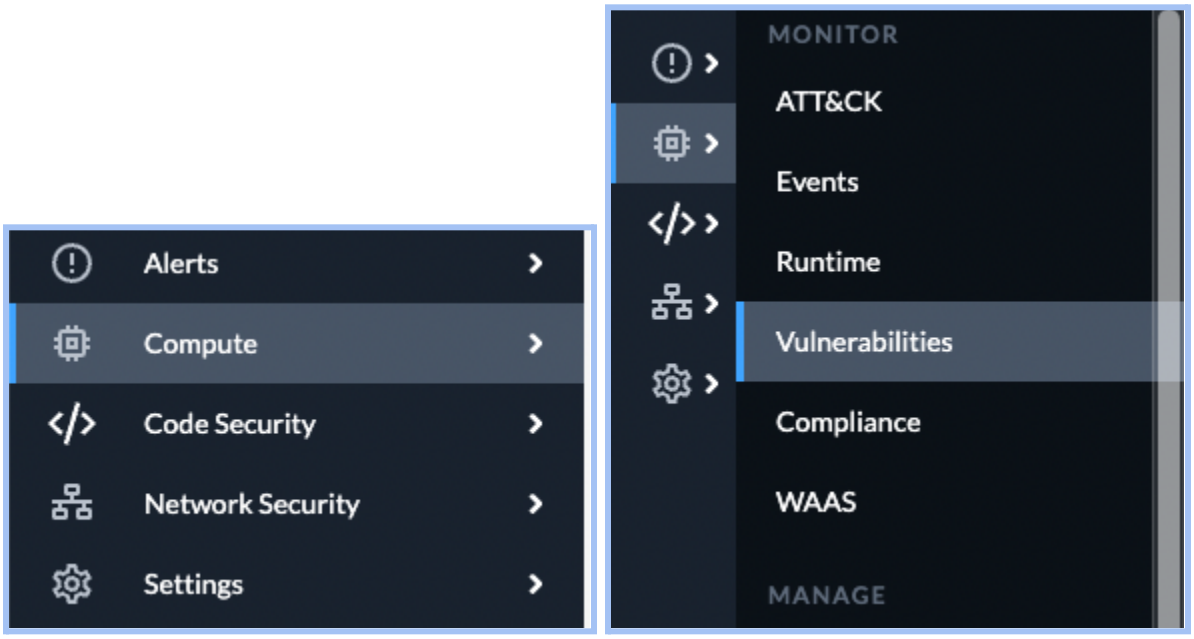
8. How many Vulnerabilities are associated with the **cryptography:38.0.1**. package? Read the **Details** section to discover more about the CVE and confirm the **Fix Version**. Are there other packages with a higher CVSS score?

   Now that you have helped to protect the Bank of Anthos banking app Software Supply Chain, let's take a look at the recent activity of the Exampli Corp developers.

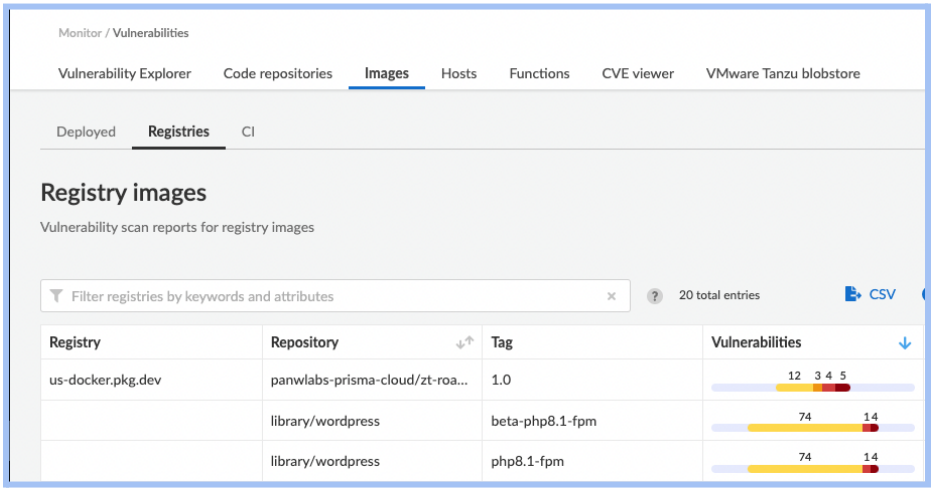
Scan Private and Open-Source Registries

Moving right in the development cycle the next step is to validate the security of the container images used in the Bank of Anthos application. Fortunately, Prisma Cloud can be leveraged to gain insights into both private and open source container registries. In this next exercise, we will examine the Bank of Anthos frontend image in Exampli Corp's private registry as well as some other open source images found in public registries.

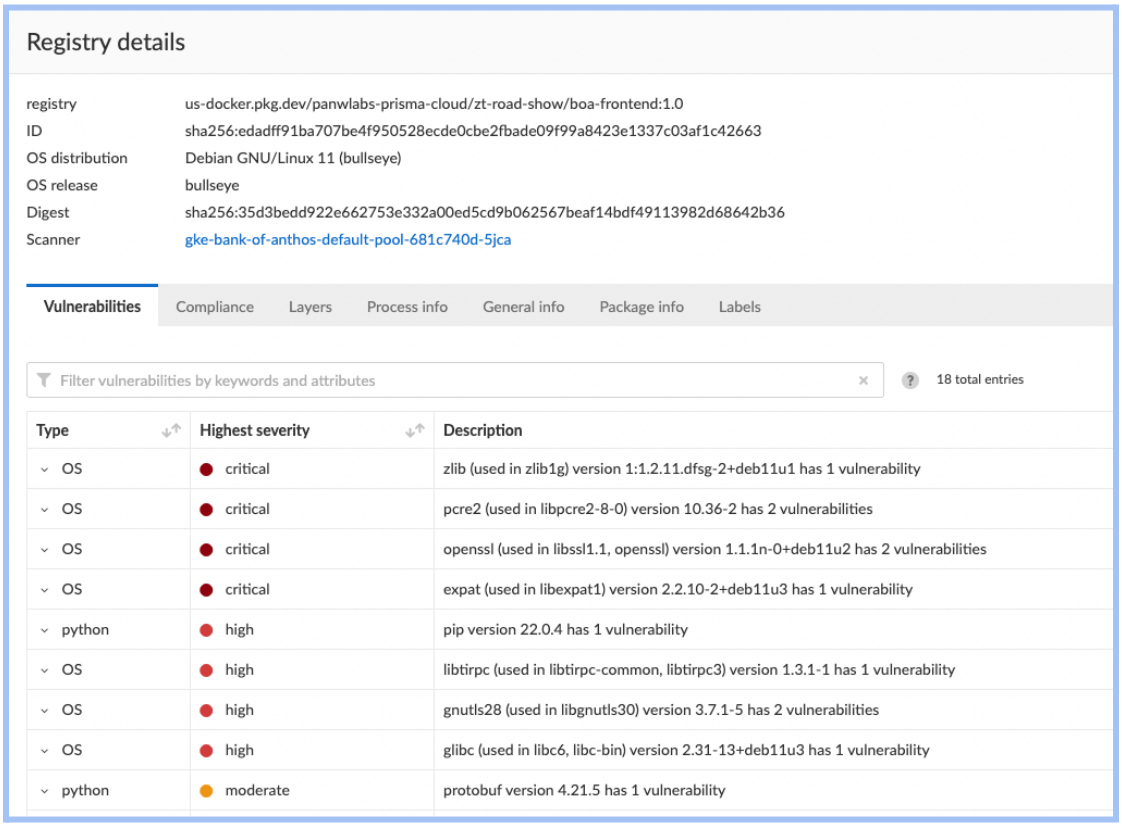
1. Use the navigation pane to select the **Compute** module and then select **Vulnerabilities** under the **Monitor** section.



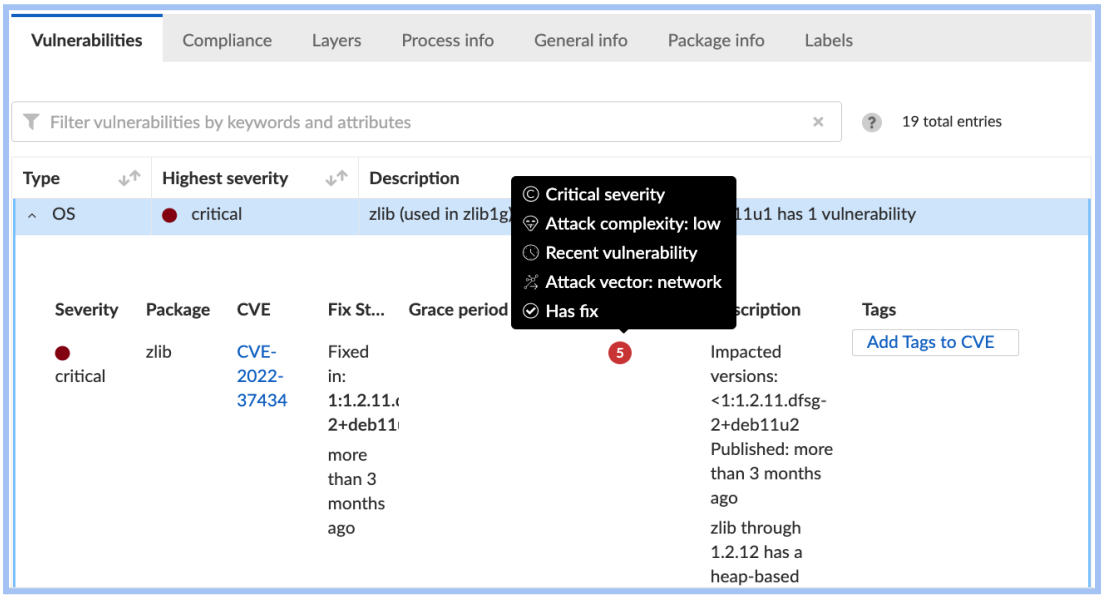
2. Click on **Images** and then **Registries** to see vulnerability scan reports for registry images.



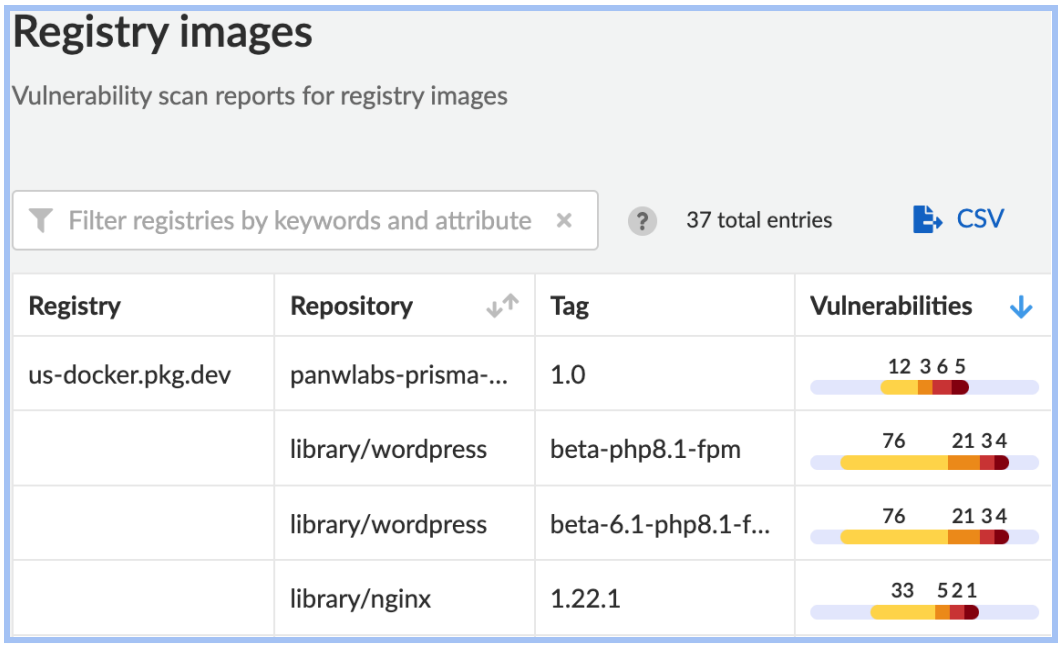
3. Here we can see reports for both public and private registries. Let's take a look at the **boa-frontend** image under the **us-docker.pkg.dev** registry being leveraged by Exampli Corp.



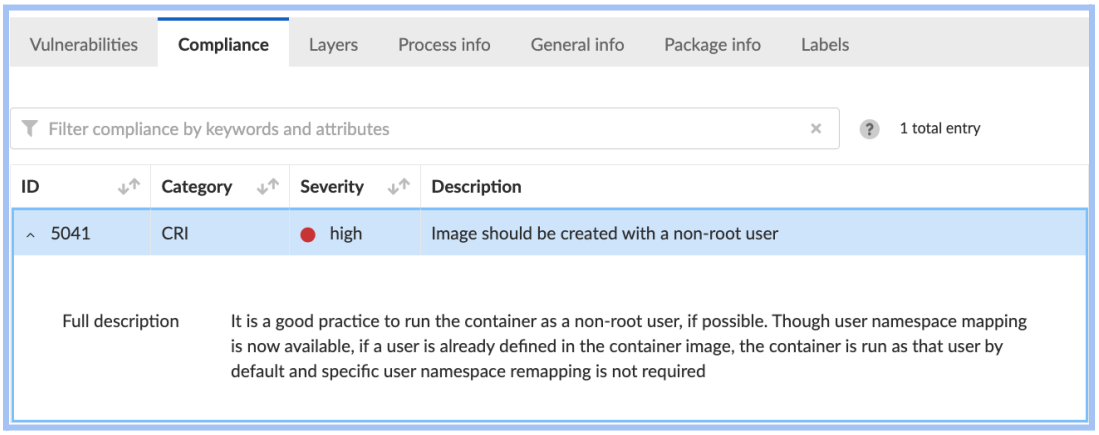
4. We can see a number of high-severity issues in the **Vulnerabilities** tab. Expand the row associated with the **zlib** package. Hover over the number circled in red under the **Risk factors** column. Here we can get an idea of attack complexity and vectors.



5. Navigate back to the **Registry images** page and look at one of the open-source package repositories. Click on the **/library/nginx** repository with the **1.22.1** tag.



6. Click on the **Compliance** tab and expand the high severity entry. Here you can gather a full description to get a better understanding of the compliance error.



7. Next, let's take a look at an open source container registry.

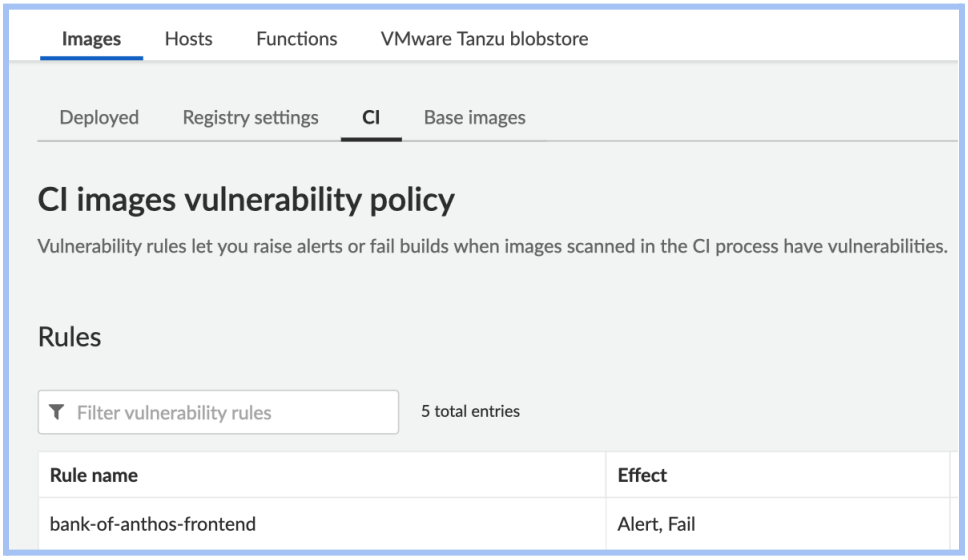
We've gotten a better idea of Exampli Corp's registry image hygiene. Let's explore further to gain more insight and protect the Bank of Anthos app with Prisma Cloud.

Deploy Secure Applications

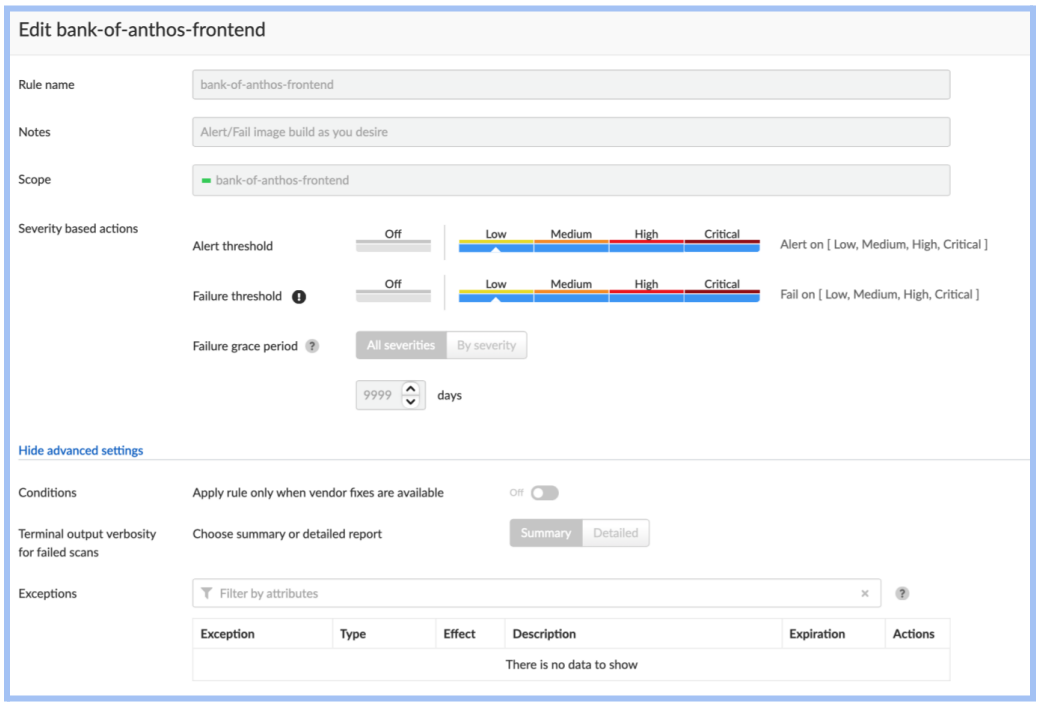
We now know Exampli Corp's images are vulnerable so we will walk through the steps on how to create a Vulnerability and Compliance rule below.

Prisma Cloud supports automation workflows with GitHub and Jenkins to integrate security in the build pipeline to scan application images, review results, and pass/fail builds based on findings.

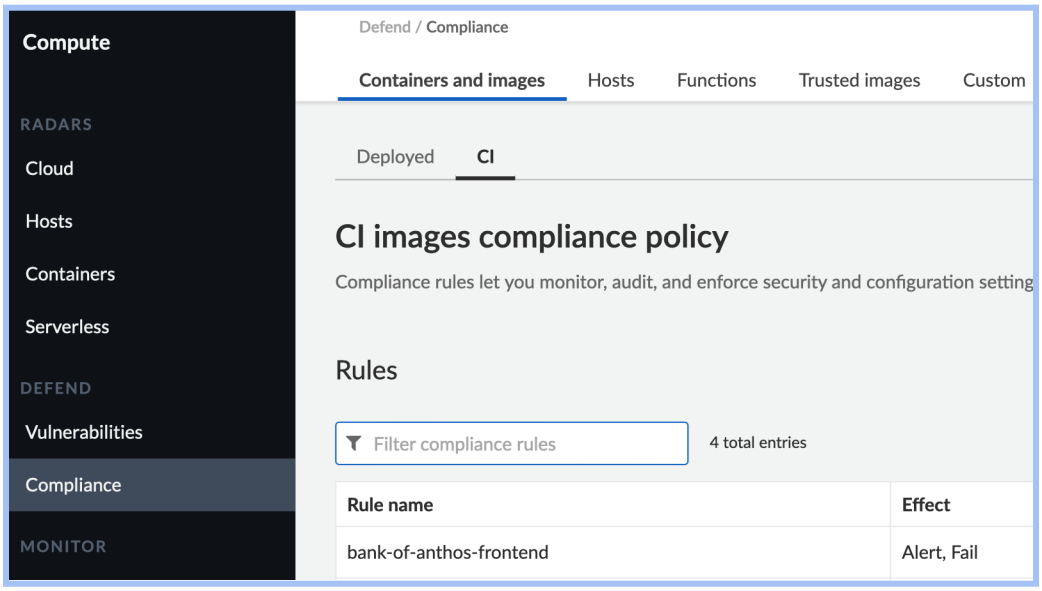
1. Authorized administrators can implement Prisma Cloud image scanning in your CI pipeline. In Prisma Cloud, vulnerability rules can be defined to raise alerts or fail builds when code repositories scanned in the CI process have vulnerabilities. View the vulnerability rule for the Bank of Anthos frontend image by going to **Compute -> Defend -> Vulnerabilities -> CI** and selecting the **bank-of-anthos-frontend** under **Rule name**.



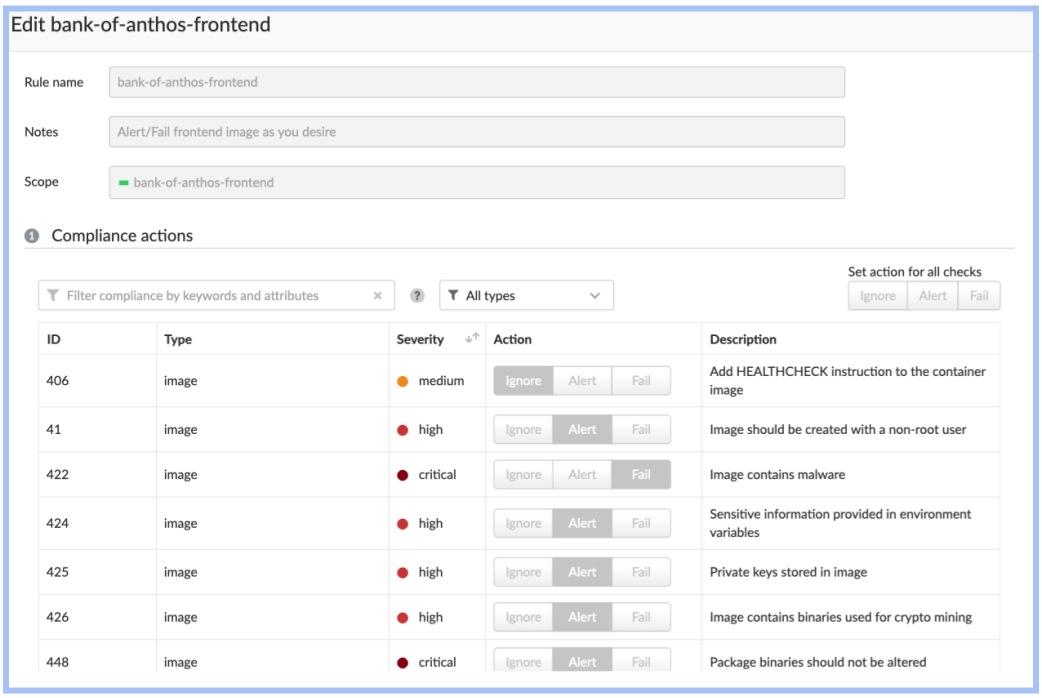
2. Here authorized administrators can edit rules to alert on and block builds based on rules they define. This can be a powerful integration to ensure that when applications hit runtime they are configured securely and that vulnerabilities are minimized.



3. In addition to Vulnerability rules, Prisma Cloud comes standard with a number of out of the box compliance policies. However, many organizations require flexibility in compliance. Prisma Cloud allows you to add custom compliance rules by navigating to **Compute** and clicking on **Compliance** under the **Defend** section.

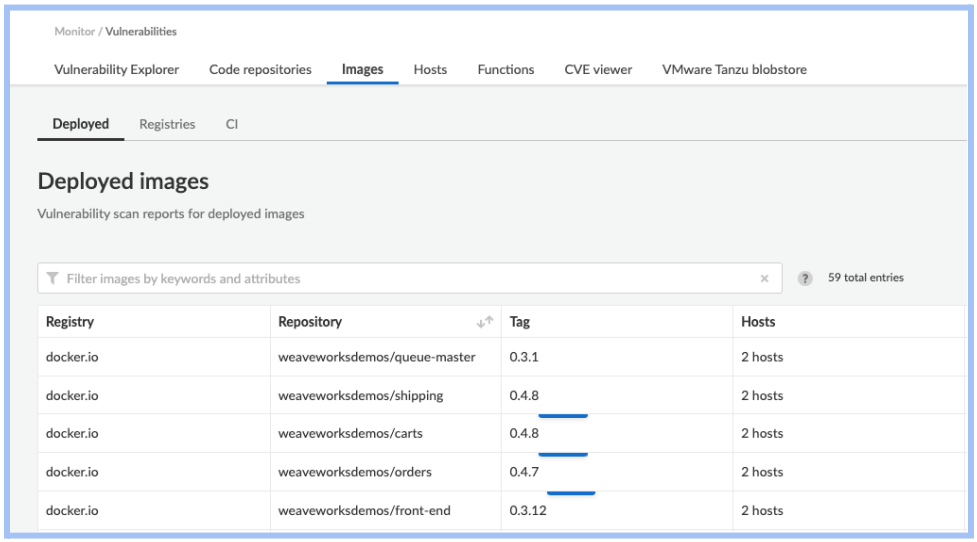


4. Here you can see all code repository compliance rules. On this page, authorized administrators are able to add and edit custom rules. To view the **bank-of-anthos-frontend** compliance rule click on the rule name.

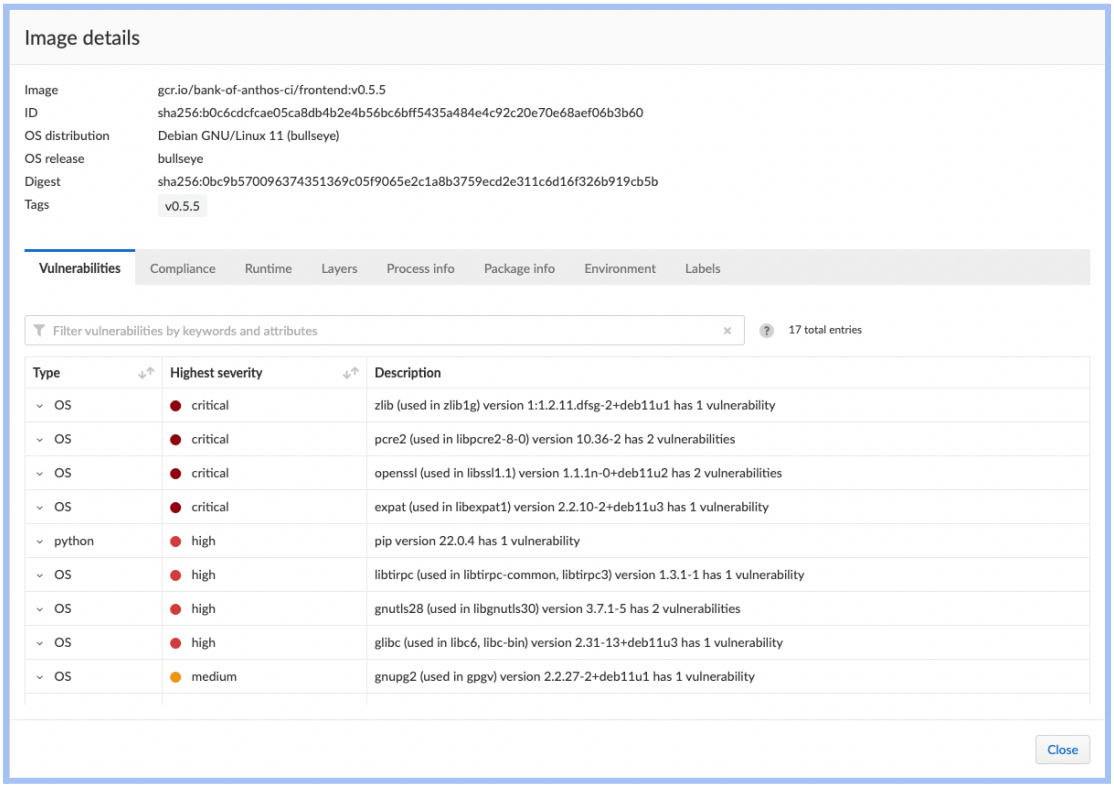


5. Here you can create custom rules that are tailored to your own business needs, standards, and organizational policies. These custom rules can help reduce the attack surface at runtime and help reduce the workload on SecOps.

6. Now, let's take a look at some of the image information about deployed images by clicking **Monitor -> Vulnerabilities -> Images -> Deployed**



7. Let's take a look at the **'bank-of-anthos-ci/frontend'** image under the **Repository** column. To view the vulnerable image simply click on the row. Your screen should look similar to the one below:



8. Explore the information found on the summary page, and answer the following questions:

   What is the base OS for this image?
   How many critical vulnerabilities are affecting this image?
   What host machine(s) is this image running on?

Securing Applications at Runtime

Wait a second, there are critical vulnerabilities in that deployed frontend image! Let's take a few minutes to investigate all GCP resources that Exampli Corp owns and secure the final phase of the application lifecycle.

Cloud-native applications allow organizations to build and run scalable applications with great agility and resilience. However, they also present unique security challenges. Ensuring applications and services are secure at runtime is a core responsibility for security teams.

The Exampli Corp team needs to take the security of the banking application just as seriously as Bank of Anthos takes the physical security of their banks. Just like physical currency, data must be secured and unauthorized access prevented when possible. In the Cloud, this means ensuring you have visibility on all of your assets, actively investigating suspicious activity, and protecting your environment from attacks. Additionally, The Exampli security team needs to ensure they follow all best practices and security frameworks that govern their industry.

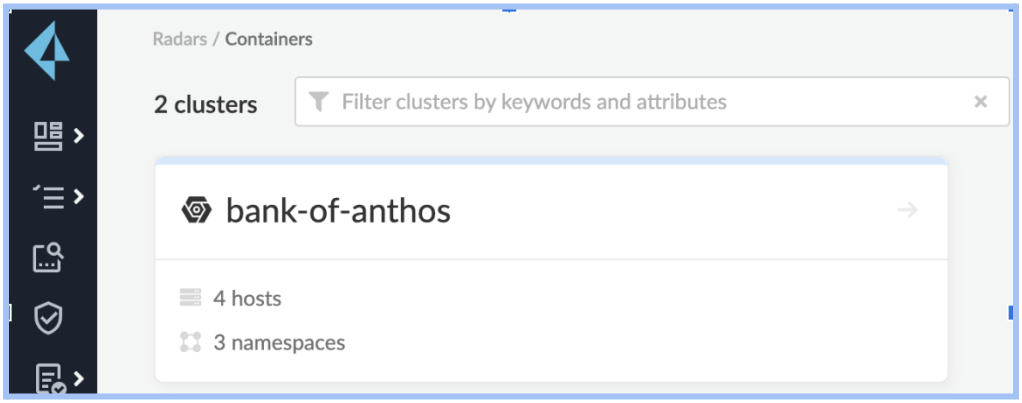
In this section, you will explore some use cases for protecting your applications during runtime.

Maintaining Visibility

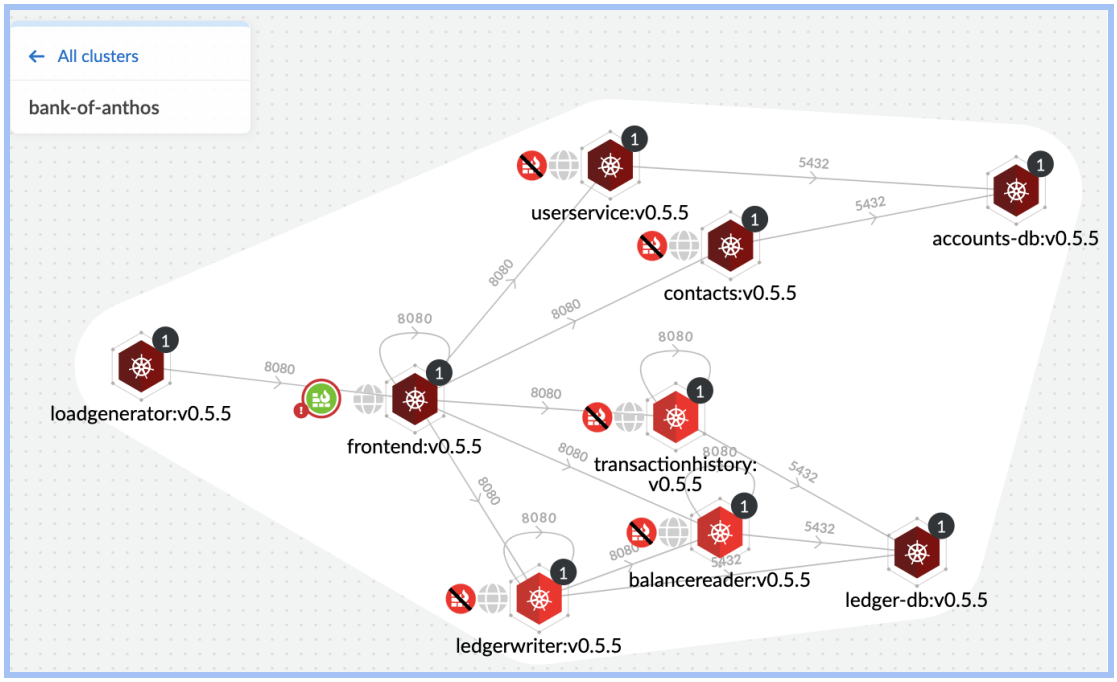
The first step to protecting applications is to gain and maintain comprehensive visibility of your applications and all their associated resources. If Exampli Corp does not have oversight of their cloud environment it is impossible for them to stop emerging threats, respond to incidents, or reduce vulnerabilities. In a modern multi-cloud and hybrid world, applications are broken into different environments across the clouds like VMs, containers, serverless compute architecture types. Maintaining visibility is essential to protecting applications at runtime.

Lets help Exampli Corp get a clear picture of the Bank of Anthos application by using Prisma Cloud's Radar feature.

1. The Radar feature lets you gain a bird's eye view to monitor and understand your cloud environment. It helps you to visualize the connectivity between microservices and search for vulnerabilities. Navigate to **Compute -> Radar -> Containers** and select the cluster **bank-of-anthos.**



2. Once you have selected the correct cluster your screen should look similar the the screenshot below:



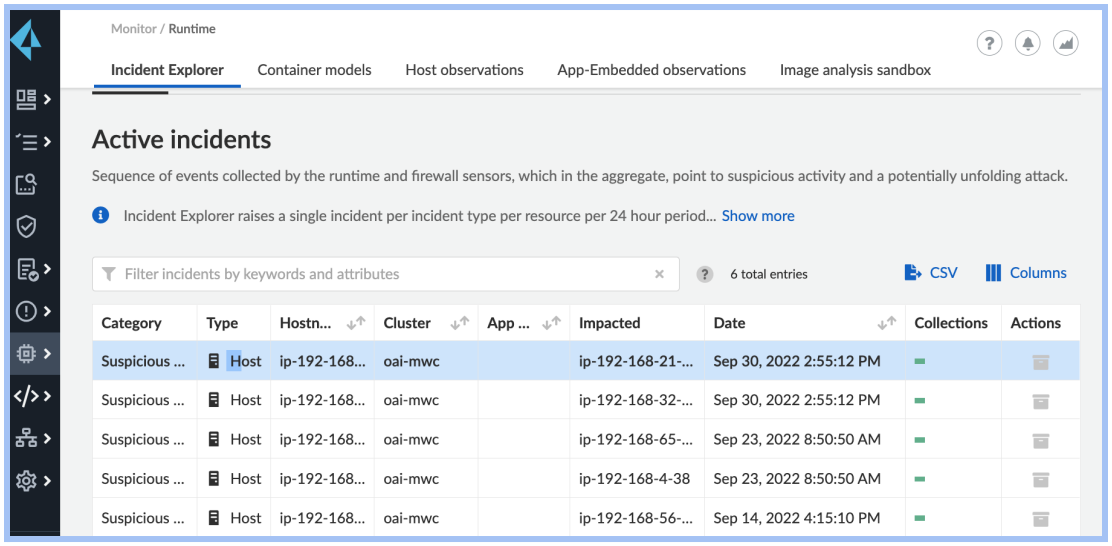
3. Radar provides a visual depiction of inter- and intra-network connections between containers, apps, and cluster services across your environment. It shows the ports associated with each connection, the direction of traffic flow, and internet accessibility. Take some time to play around with the Radar feature and think through the following questions:

What type of information can you learn by clicking on the nodes?
How does this visibility help you protect your applications?

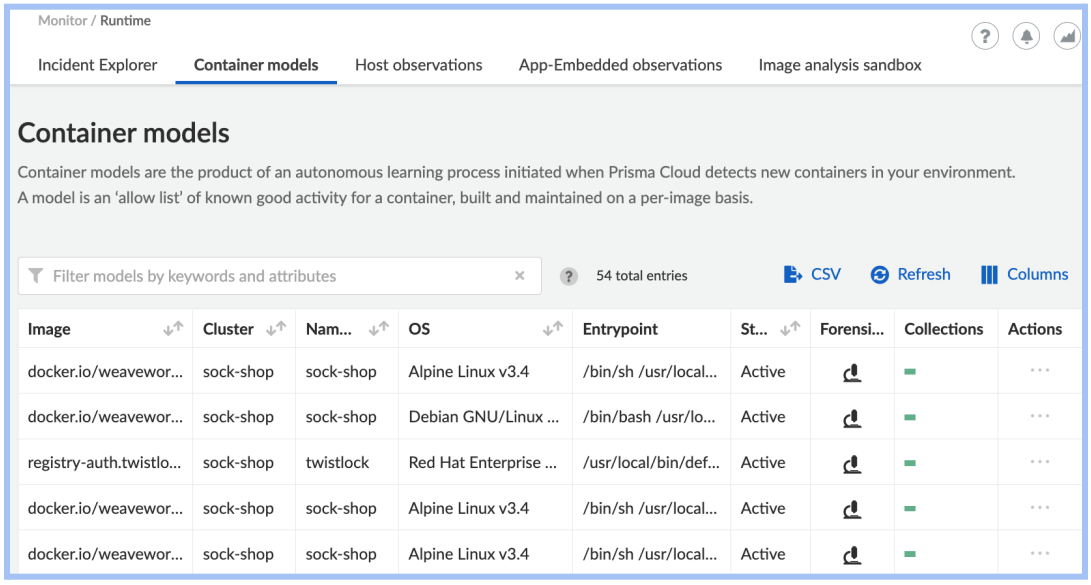
Investigate Incidents at Runtime

Exampli Corp has adopted Prisma Cloud and maintained consistent visibility on their applications. But seeing your resources does not make them immune to incidents during runtime. Thankfully, Prisma Cloud offers AppSec solutions for the entire lifecycle of your applications. By using incident explorer the Exampli Corp Security team gets real-time detection and analysis of all potential threats that violate their security policies. Let's take a look and see if there are any incidents in our cloud environment.

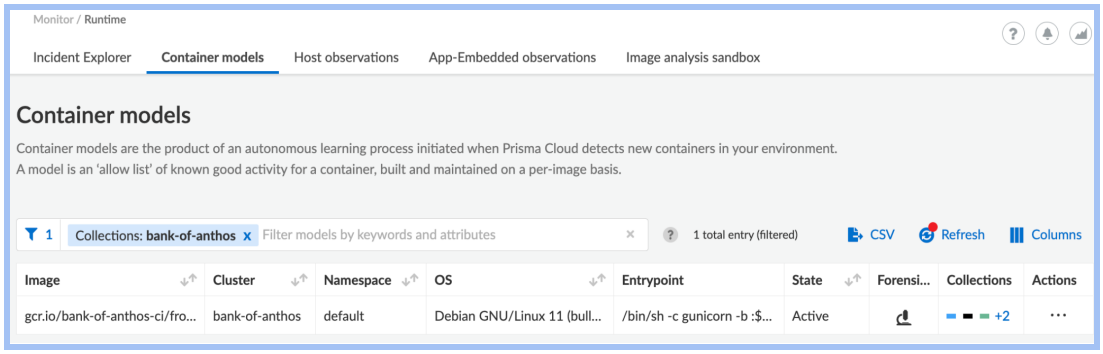
1. Let's begin by navigating to **Monitor -> Runtime -> Incident Explorer.** On this screen we can view suspicious events collected by our runtime and firewall sensors. Take a look around the page and get a feel for the types of incidents being reported.



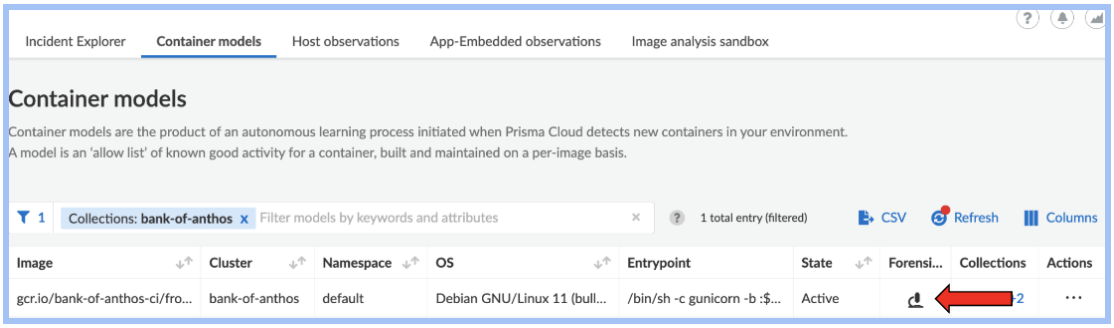
2. Next, click on **Container Models** tab. Here we can view all of our containers and take a deeper look at the forensics to see if there are any security concerns that slipped through the cracks.



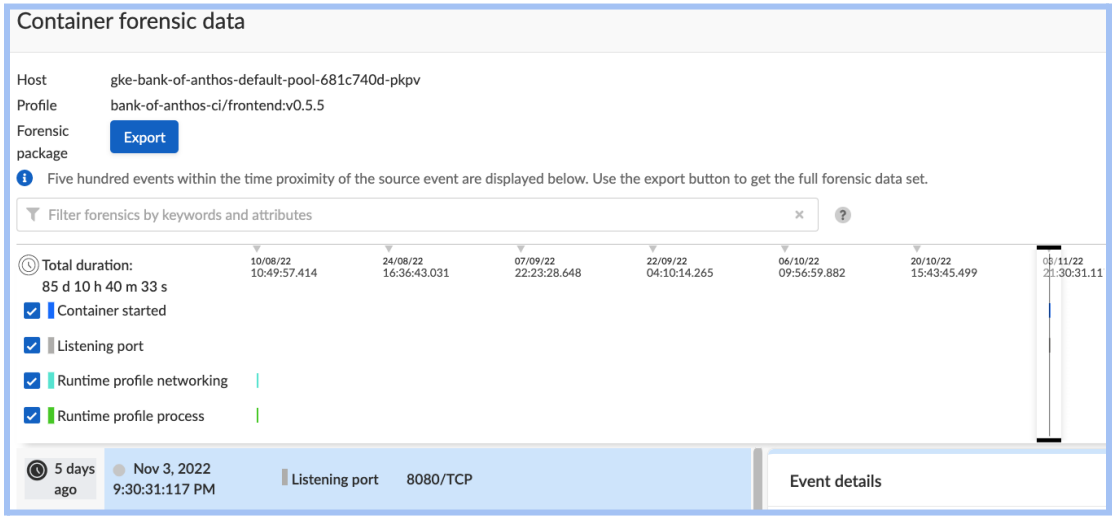
3. We know that Exampli Corp is getting ready to release the new Bank of Anthos App, so let's search "**gcr.io/bank-of-anthos-ci/frontend:v0.5.5**" in the filter box to see how the containers are looking.



4. Great! Now let's click on the microscope icon to take a look at the digital forensics



5. Here you can see all the events displayed. Let's take a deeper look at these events to get some details and see what has been happening with our bank-of-anthos-frontend.



6. Spend some time looking through the events and answer the following questions:

What information can be gathered from the events?
What types of events are being displayed?
Should we be concerned about any of these events? What action if any should be taken?

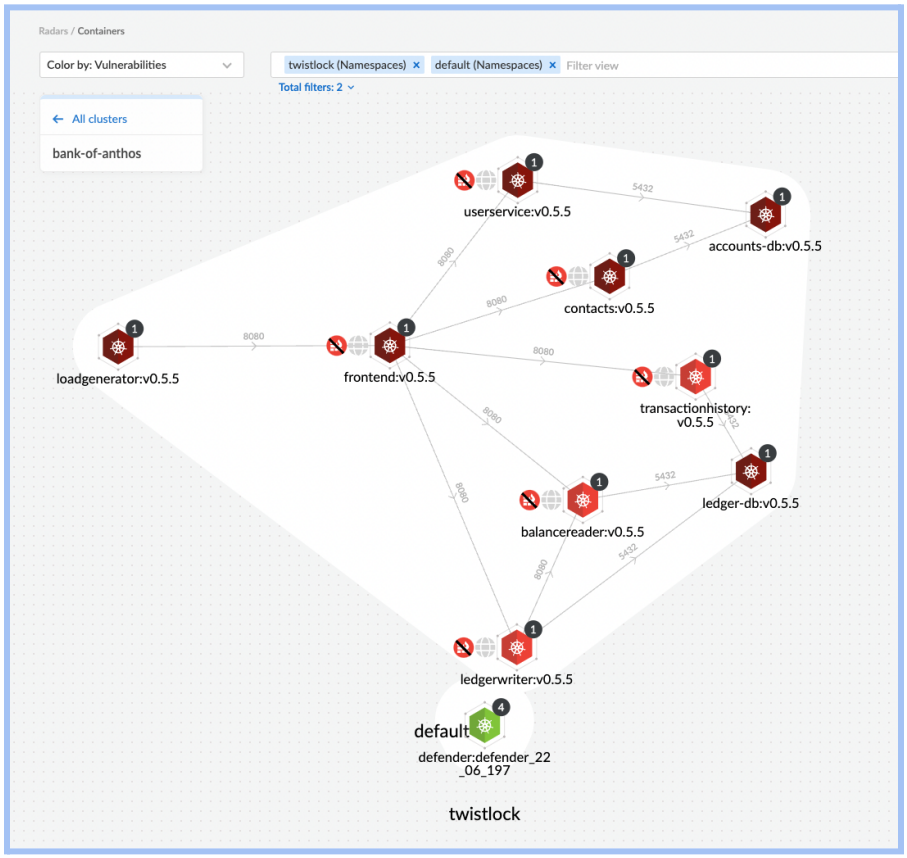
Preventing attacks

One of the most important parts of securing the banking app during runtime is preventing attacks. Bank of Anthos invests significant resources in guaranteeing the security of their physical banks. They have security cameras to log and monitor access, sensors to alert on threats, but most importantly, banks have safes and security guards. In the worst case outcome if all prior controls fail there are physical preventative capabilities to protect the critical assets.

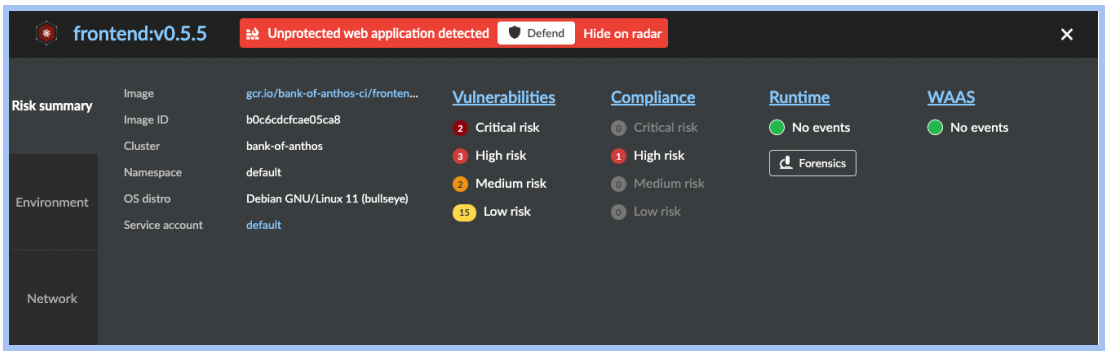
Prisma Cloud provides process level detail and critical forensic information to give both visibility and aid in threat investigations for indicators of compromise. Spend some time reviewing the forensic information Prisma Cloud provides, feel free to look back in time.

1. Next, let's take a closer look at some containers that are running on those GKE hosts we were just looking at. Navigate back to **Radar** -> **Containers**

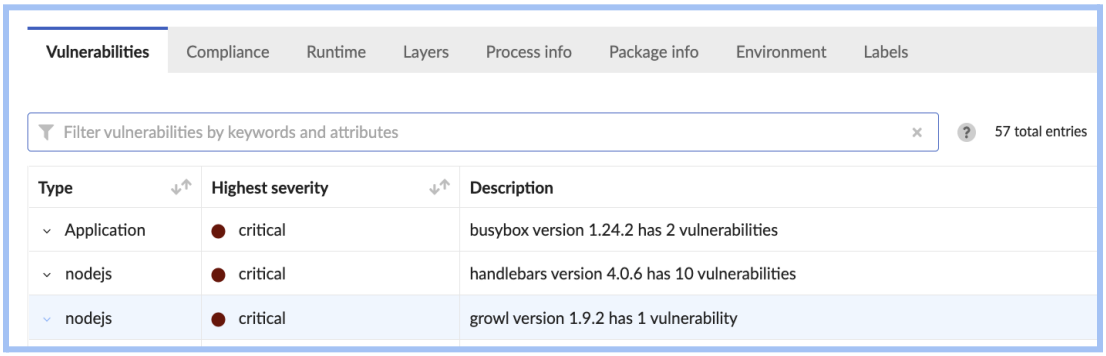
2. Let's look at the default namespace for the bank-of-anthos cluster. Here you can see the microservices running in containers that are powering this simple application. Your screen should look similar to the screenshot below :



3. Let's dive deeper into the '**frontend:v0.5.5**' microservice. Click on the associated container. Your screen should look similar to the screenshot below:

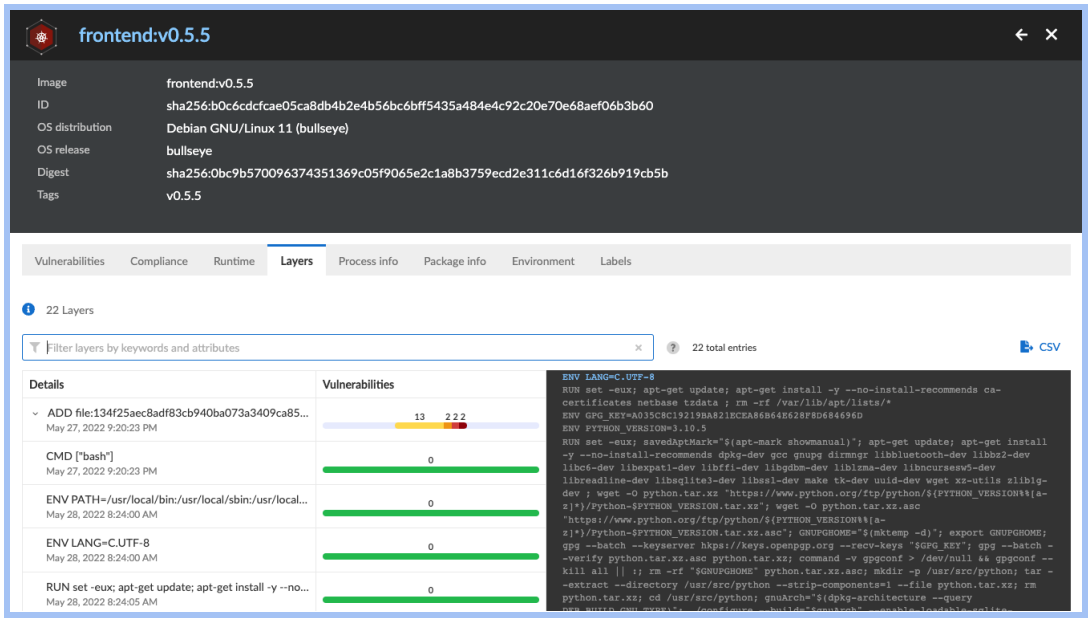


4. Take some time to review the **Vulnerabilities** page.



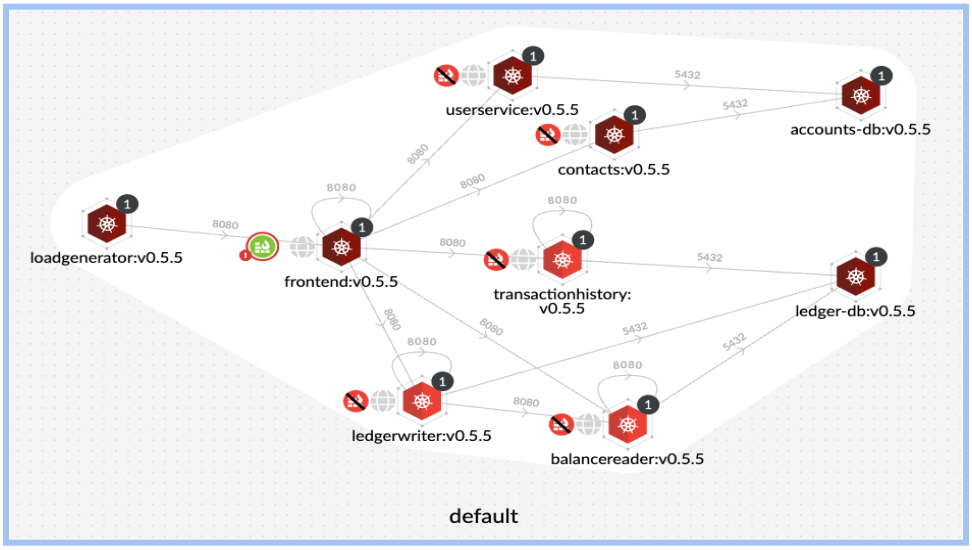
5. Of the identified OS vulnerabilities, which one has the highest CVE? Do all the identified vulnerabilities contain a fix?

6. Take a look at the **Layers** tab to view the dockerfile that built this image and find where vulnerabilities were introduced.



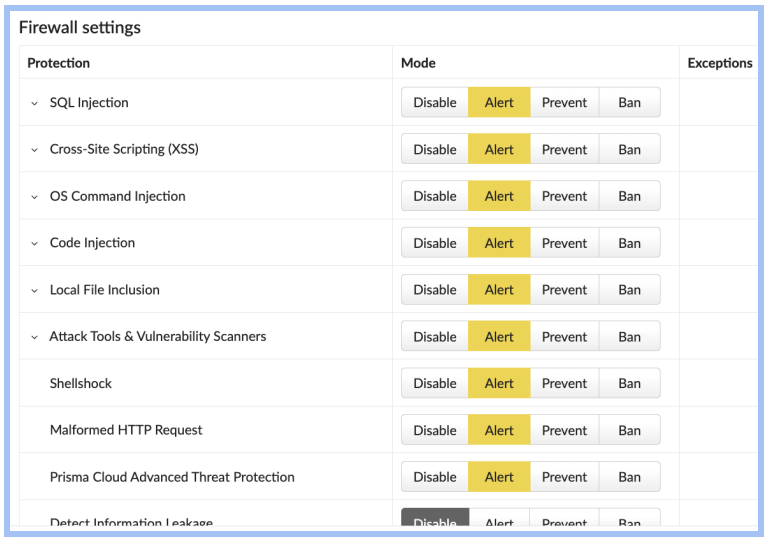
7. In addition to image scanning, runtime visibility and protection; the Prisma Cloud defender also provides Web Application Firewall and API Security

8. To take a look let's navigate back to **Radars -> Containers** and ensure you are viewing the **bank-of-anthos** app.



9. The green firewall log indicates that the front-end microservice is protected by Prisma Cloud.

10. Administrators can define rules to provide web application and api security capabilities to protect web applications. Prisma Cloud supports VM Hosts, Containers, Application Embedded, and function deployment architectures.

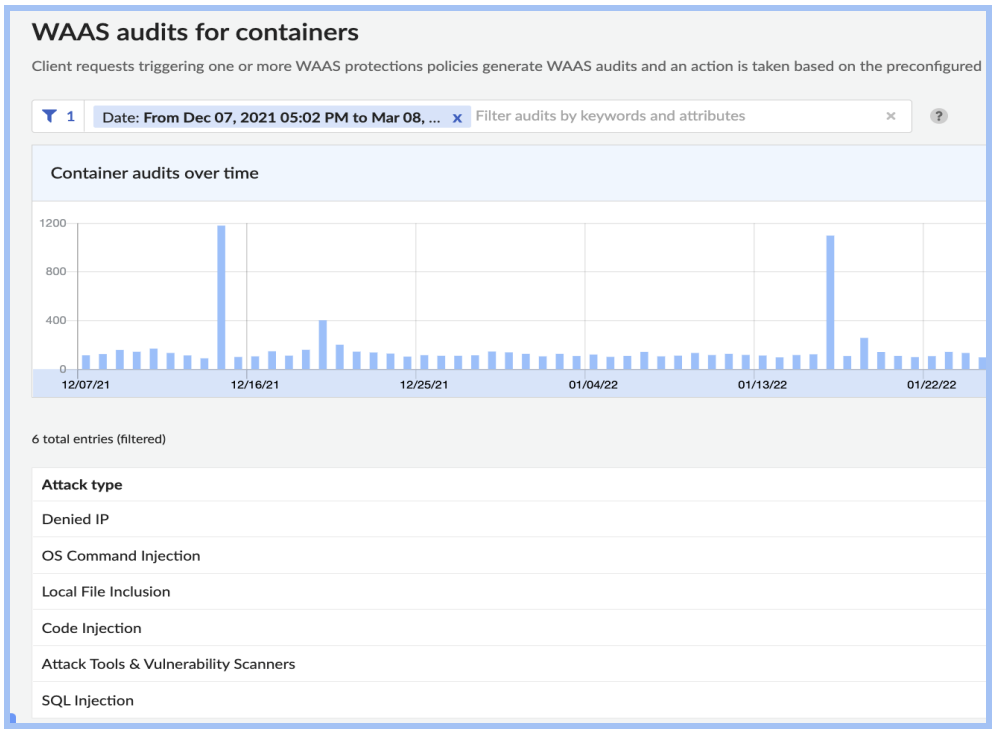


11. Administrators can define Custom Rules that provide Virtual Patching capabilities to protect against attacks exploiting CVE's that have not yet been patched. For example check out the log4j blog where you can find more information about custom rules that were created to protect our customers. (Link)

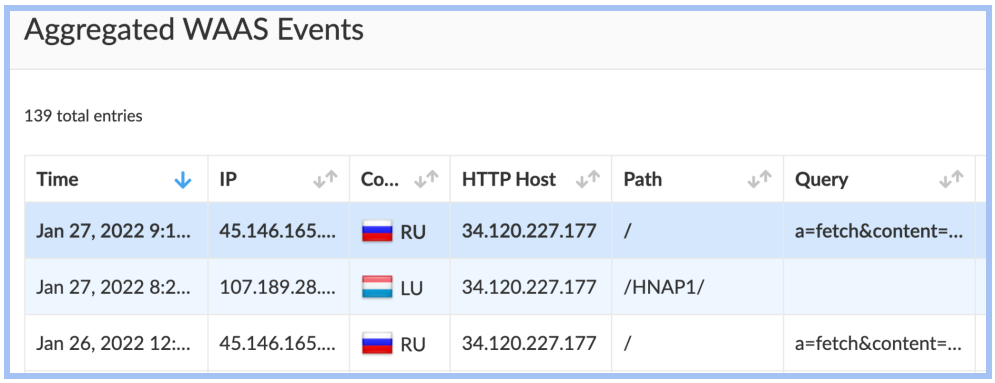
12. Take a look at some of the attacks the defender has prevented by navigating to **Monitor -> Events -> WAAS for Containers**.

    Thankfully no major events have occurred yet on the *accidentally* deployed banking app, however we know Exampli Corp has a bad track record for not securing their apps. Let's look at some historical data from last holiday season.

    Click in the filter bar and select **Date**. Enter Dec 1, 2021 for the **From** date and Jan 31, 2022 for the **To** date.



13. Let's dive into one of the **OS Command Injection** attacks. Click the **total** value to view the events.



14. Select one of the entries and answer the questions below:

    What was the result of the attack?
    Was it blocked?
    What was the http method that was used in the attack?
    What container image was attacked?

Summary \ Resources

The Prisma Cloud team here at Palo Alto sincerely hopes you enjoyed this workshop. Today, organizations need a growing set of capabilities to secure modern cloud applications. Implementing AppSec into your development and security workflows with Prisma Cloud can provide the increased speed and agility your organization needs to implement zero trust throughout the entire development lifecycle. Check out the resources below for more information!

Live Workshops

DevSecTalks Podcast

Supply Chain Security

Cloud DevSecOps

Learn More

Sample Git Repositroies

TerraGoat - Vulnerable by design Terraform Infrastructure
Cfngoat - Vulnerable by design Cloudformation Template
CdkGoat - Vulnerable by design AWS CDK Infrastructure
BicepGoat - Vulnerable by design Bicep and ARM Infrastructure
KubernetesGoat - Vulnerable by design Kubernetes Cluster
KustomizeGoat - Vulnerable by design Kustomize deployment
SupplyGoat - Vulnerable by design SCA