Implementing Zero Trust Within Workload Communications

v0.1, October 2022, Cloud Connector Overview and Protecting Cloud Workloads with ZIA

Introduction

Welcome to the Building Zero Trust Workload Communications lab!

Keeping applications and cloud workloads secure is a growing concern across all industries. Zero trust architecture is an effective approach to workload security. Zscaler Workload Communications on Amazon Web Services (AWS) allows you to securely connect your applications anywhere by minimizing the attack surface, preventing lateral movement, and reducing the risk of bad actors gaining access to your data. Unlike legacy firewalls and security appliances, Zscaler Workload Communications directly connects any application to any destination and enforces least-privileged policies for zero trust security.

This workshop covers two main areas:

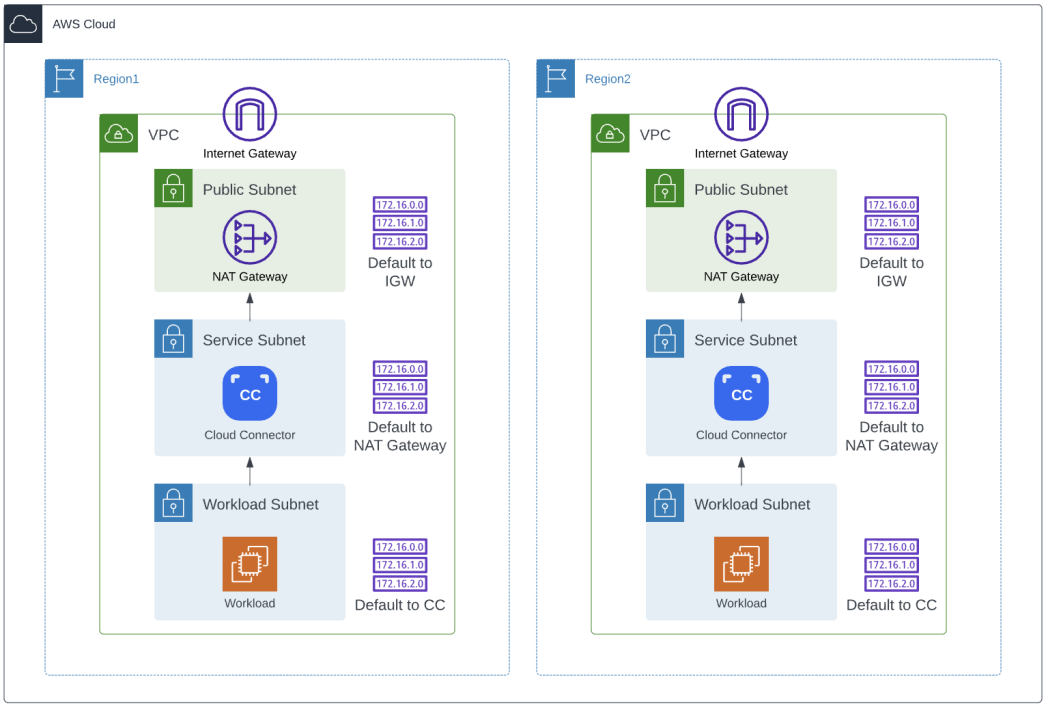
- Cloud Connector Overview
- Protecting Workload Internet Access with ZIA

Prerequisites

Your pod has been preconfigured with Cloud Connector appliances. As such, deployment of Cloud Connector appliances is not covered within this lab. For more information on provisioning and deploying Cloud Connector appliances, please refer to the reference documentation here. We strongly recommend reviewing this workflow prior to proceeding so that you have a complete understanding of the remaining tasks.

Topology Diagram

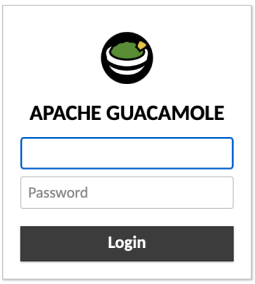
This workshop deploys resources in the AWS us-west-1 (Region1) and us-east-1 (Region2) regions:



Accessing the Environment

Your pod has been configured with a Bastion host running Apache Guacamole. Guacamole is an HTML5 web interface that can be used to proxy communication between your machine and each of the lab components you wish to access without the need for a VPN or client.

The event coordinators will send you a link to access your pod. Simply clicking on the URL and it should direct you to your pod's sign-in screen:

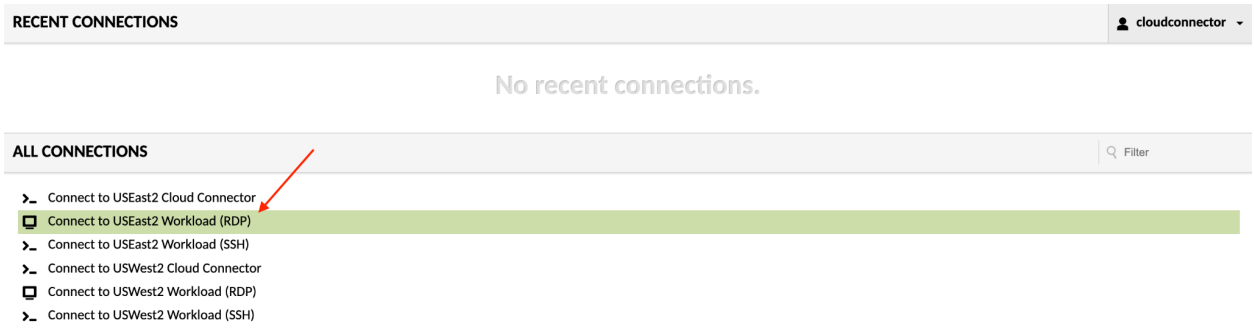


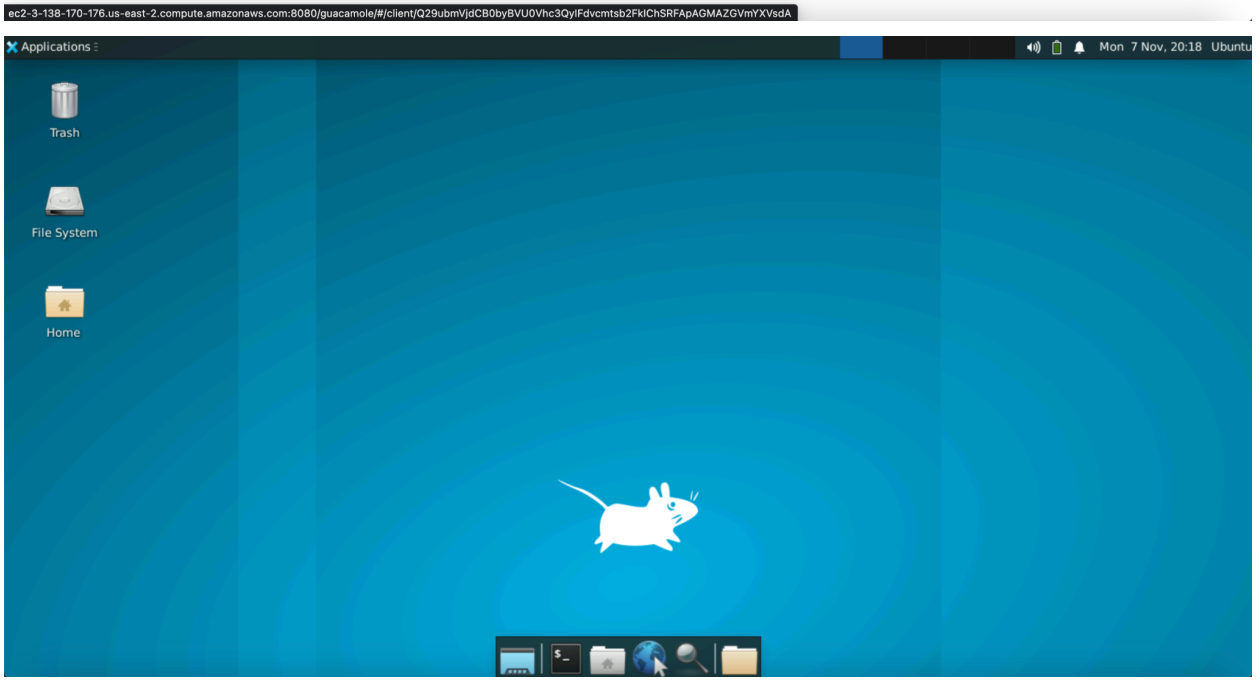
Sign in using the Guacamole username and password provided:

Guacamole **Username**: cloudconnector
Guacamole **Password**: CloudConnector2022!

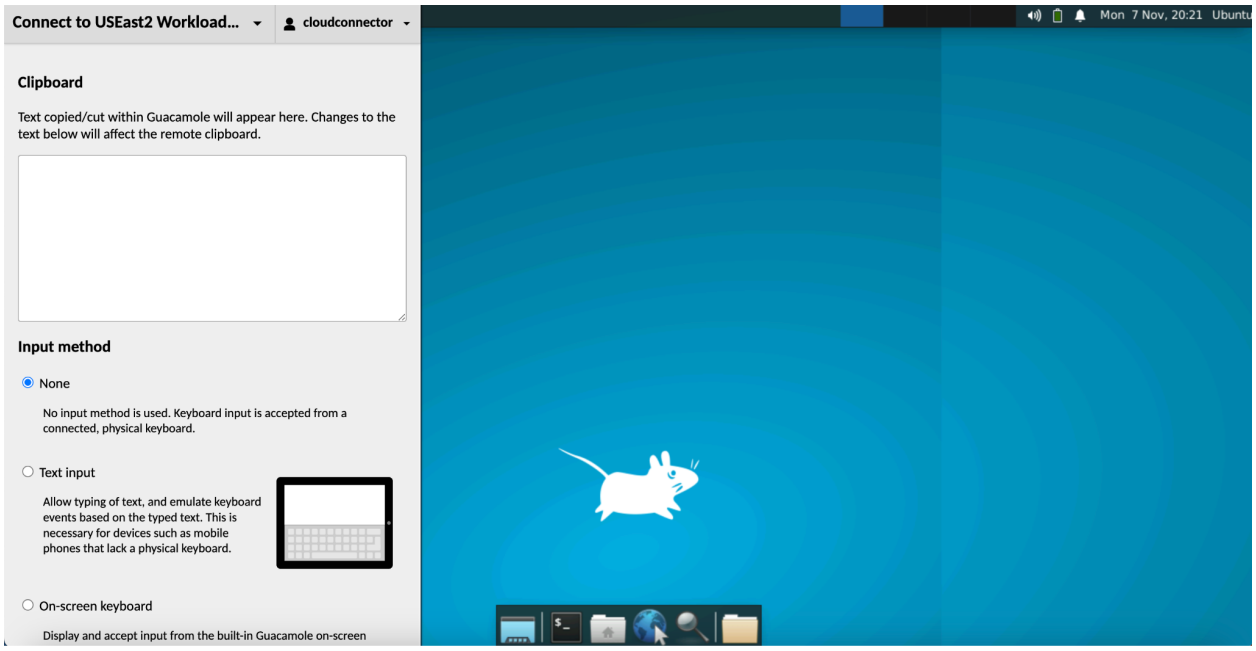
Once in, notice that several hyperlinks exist for accessing each of the pod's devices: Region1 Workload (RDP and SSH), Region2 Workload (RDP and SSH), Region1 Cloud Connector (SSH), and Region2 Cloud Connector (SSH):

Clicking on the hyperlinks will redirect you to the console of the corresponding device, where you can complete your lab tasks:





To return home, simply press the back button on your browser. To adjust settings within your current window, press CTRL+ALT+SHIFT (CTRL+OPTION+SHIFT on Mac). The latter option (shown below) is useful for split-screening consoles, copying or pasting, and quickly navigating between connections:



Module 1: Cloud Connector Overview

Zscaler Cloud Connector is a virtual machine (VM) that simplifies traffic being forwarded to Zscaler services. It extends the capabilities of Zscaler Internet Access (ZIA) and Zscaler Private Access (ZPA) to cloud-native workloads, which allows enterprises to secure cloud workload communications over any network. The remainder of this lab will focus on Cloud Connector with ZIA, but additional labs will focus on ZPA as well.

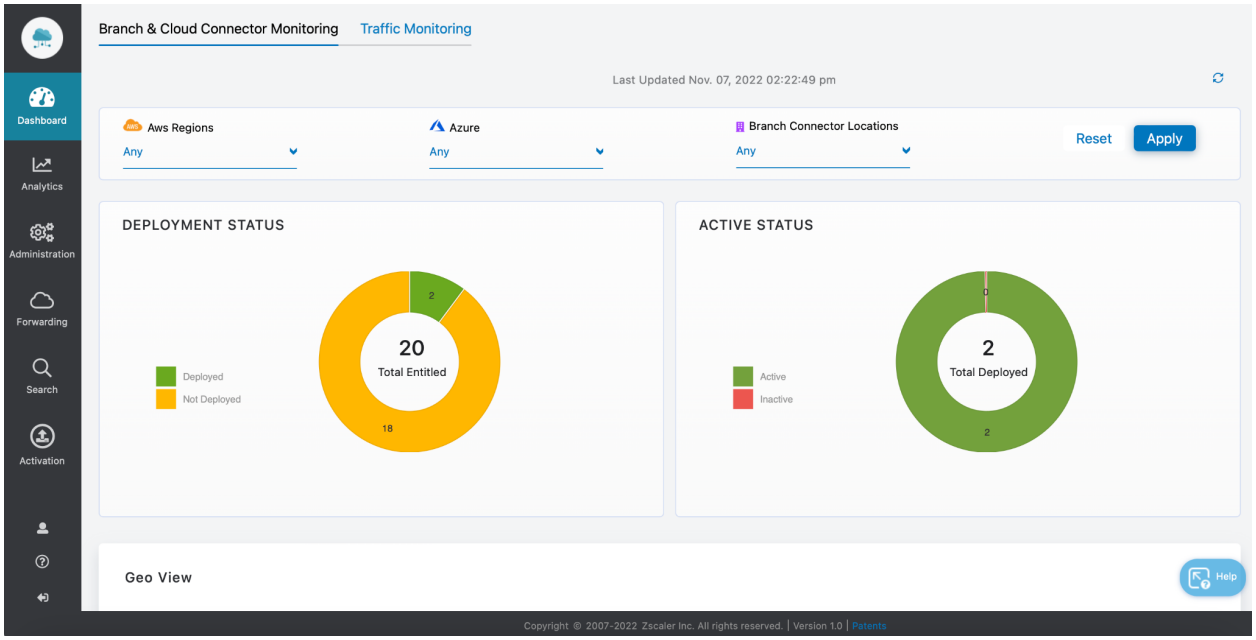
Dashboard and Navigation

For this first module, navigate to the Cloud Connector dashboard URL found in your registration email. The Cloud Connector dashboard has three primary functions:

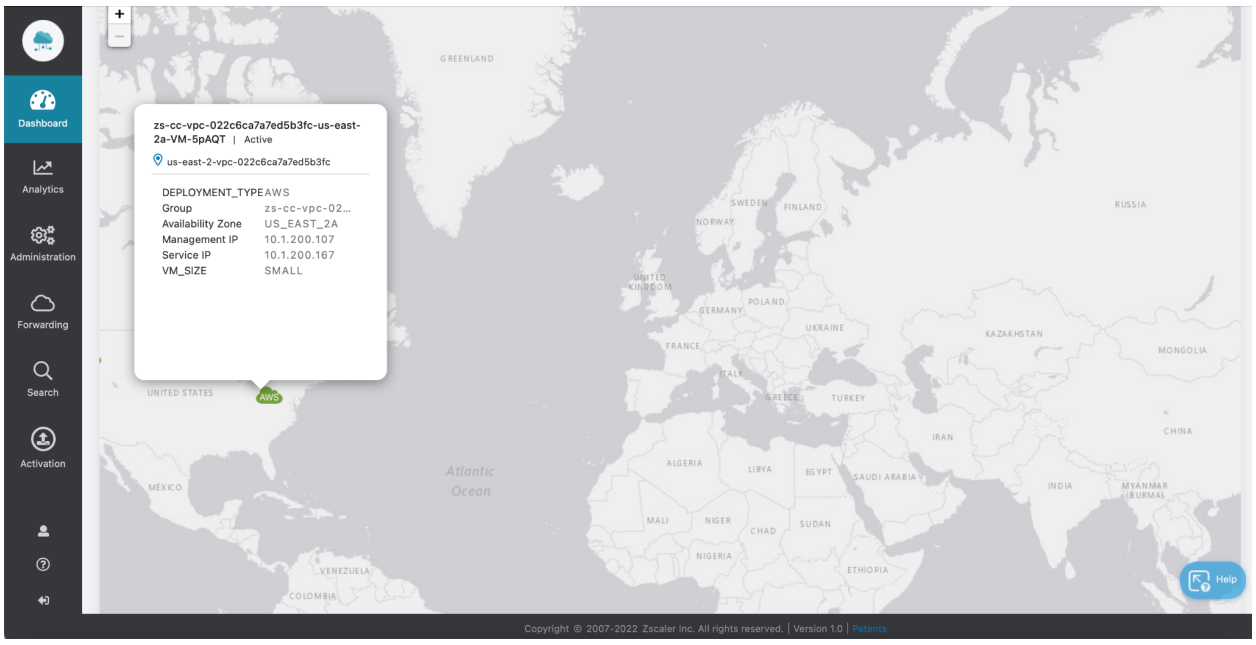
- Facilitating the onboarding, provisioning, and registration process of appliances within various Branch and Cloud environments
- Providing a new logging perspective on traffic leaving or transiting the cloud. This can aid in troubleshooting as well as threat correlation.
- Providing a traffic steering mechanism to allow administrators to granularly control how traffic is forwarded through the appliance (ZIA, ZPA, or direct).

Log in to the Cloud Connector portal using the credentials provided in the event presentation:

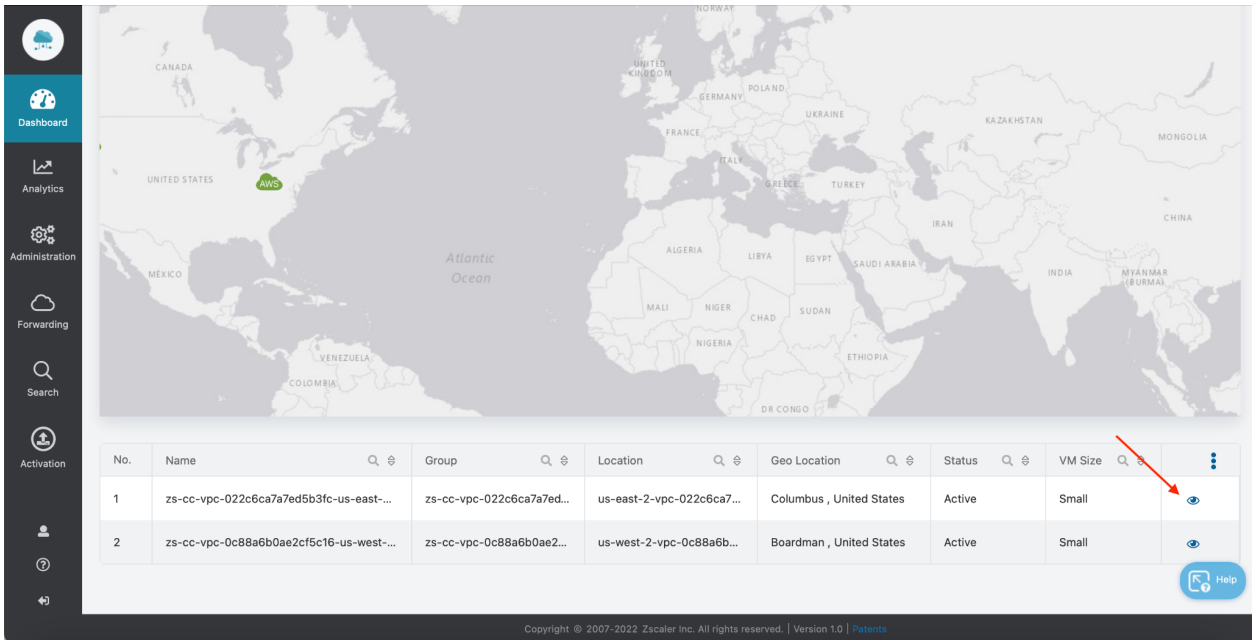
Cloud Connector portal **Username**: student@relacszXX.net (where XX is your pod #)
Cloud Connector portal **Password**: 7f#1CQ^DU03K



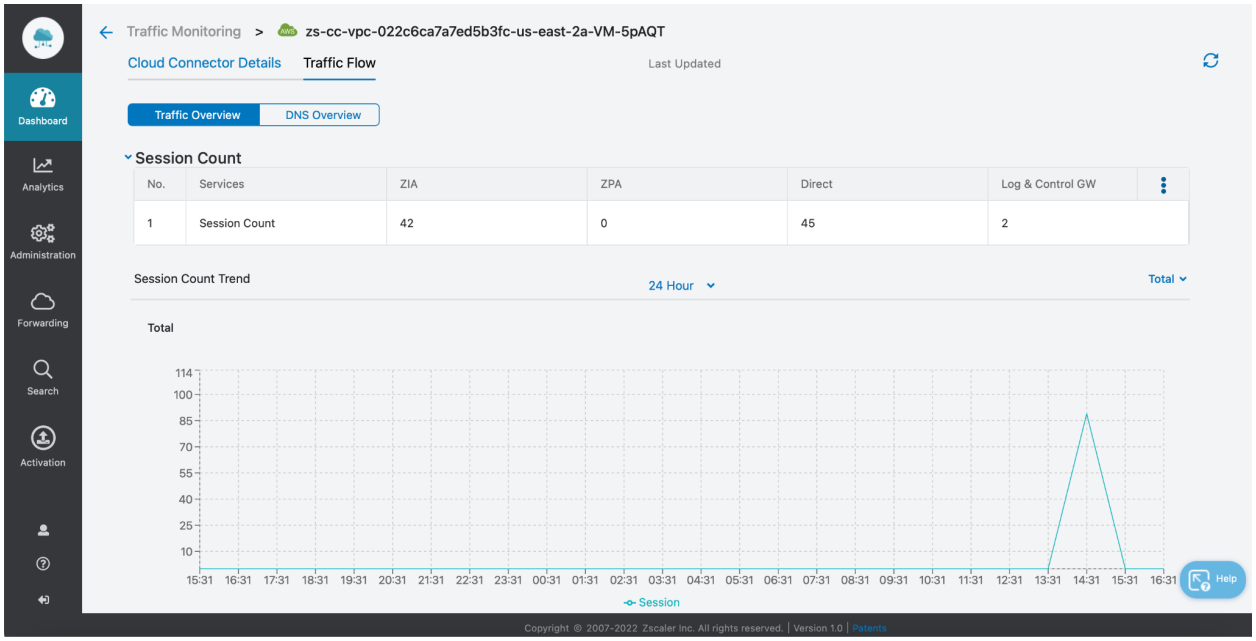
Once logged in, you're presented with an overview dashboard of your environment. The dashboard is broken into three sections. At the top, you'll see the number of Cloud Connector appliances deployed and the number of active vs. inactive. Scrolling down, you'll notice a geographical view of where each of the Cloud Connector appliances are deployed. Click on the Cloud Connector icon on the map to view additional information about the appliance. Here, you'll see the appliance size, health, status, VPC information, etc.:



Scrolling down further, you'll see a listing of the Cloud Connector appliances. Click on the view details icon (eyeball) to the right of one of your Cloud Connectors:



In this screen, you can view additional information about the Cloud Connector appliance, such as General and Management Information and Forwarding Information. In the upper left portion of the screen, click the Traffic Monitoring tab:



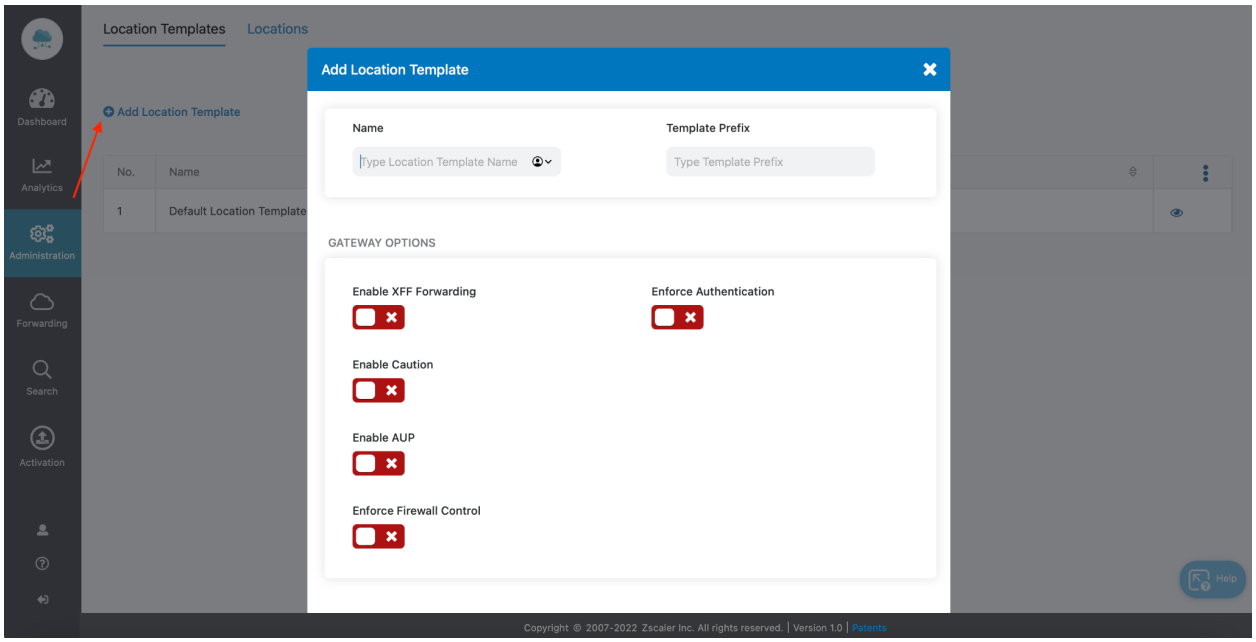
Here, we can see statistics about the traffic that this appliance is processing. Review the output and proceed to the Provisioning and Template Management section.

Provisioning and Template Management

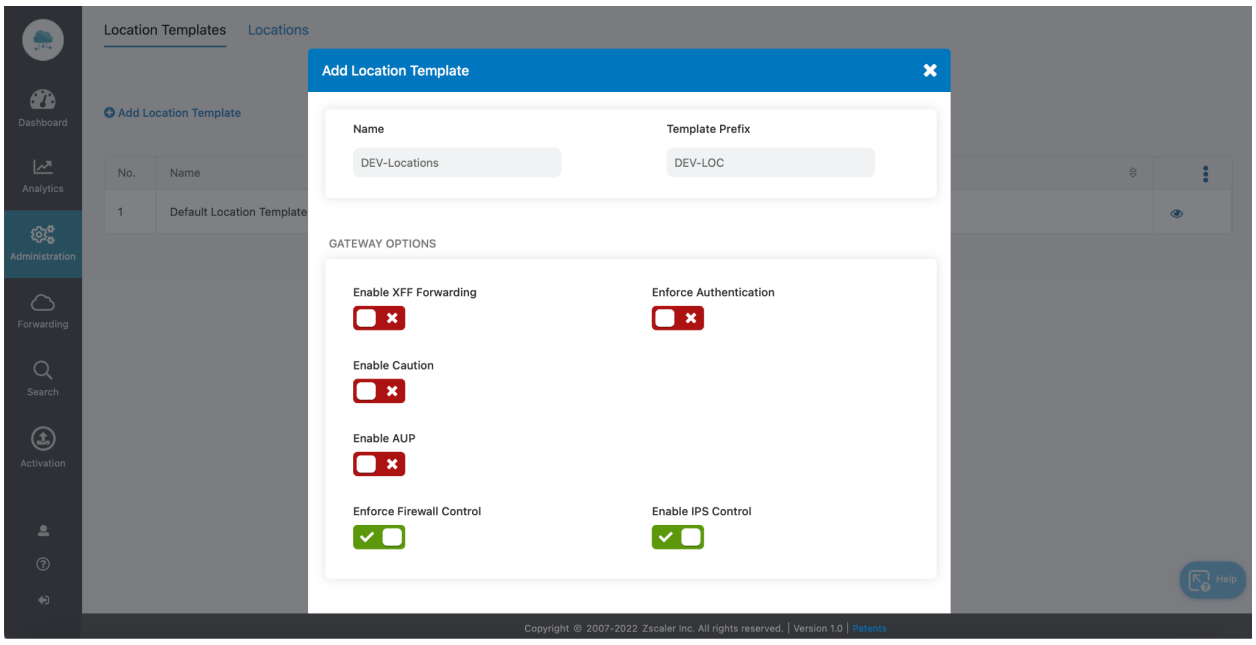
Cloud Connector appliances use Provisioning Templates and Location Templates to bootstrap themselves when they register. Navigate to Location Templates inside the Administration menu. Note that there is a Default Location Template already present. Zscaler Cloud Connector

appliances automatically create Locations based on the Cloud Service Provider networks that they serve. Controlling which features are enabled or disabled for dynamically created Locations is the job of a Location Template.

Click the Add Location Template button:



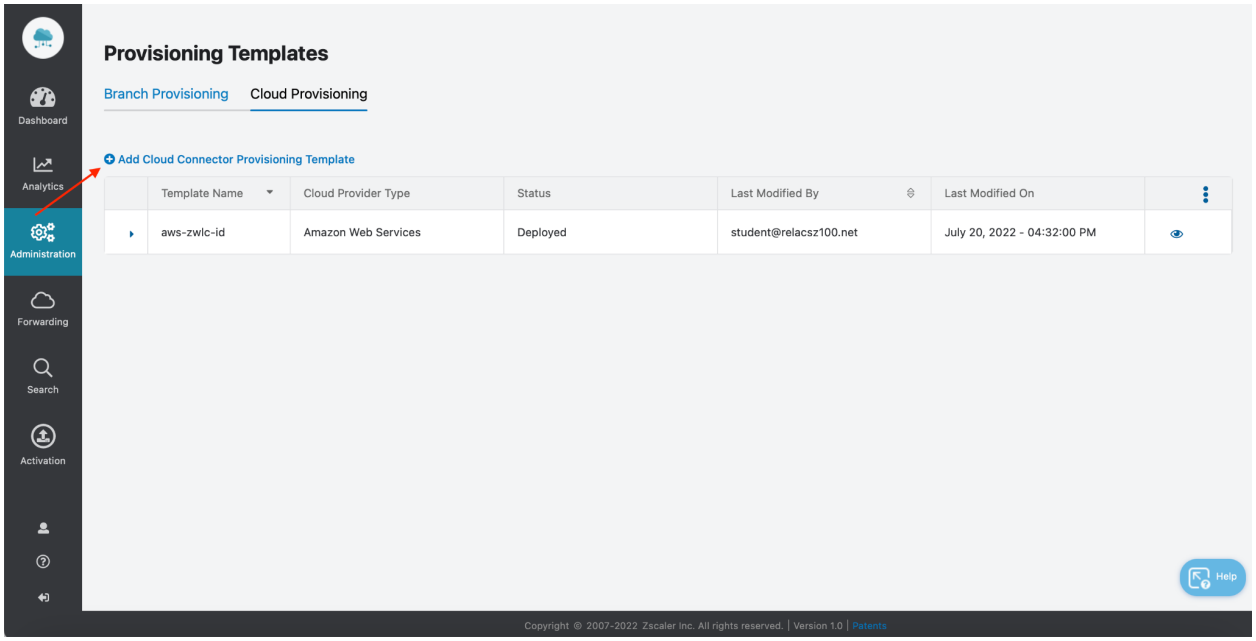
Provide a name and, optionally, a Template Prefix. The Template Prefix will be prepended to all Locations this template is attached to in order to help make a Location more easily identifiable. Select the options you wish to enable and click the Save button (this template will not be used, so feel free to explore):



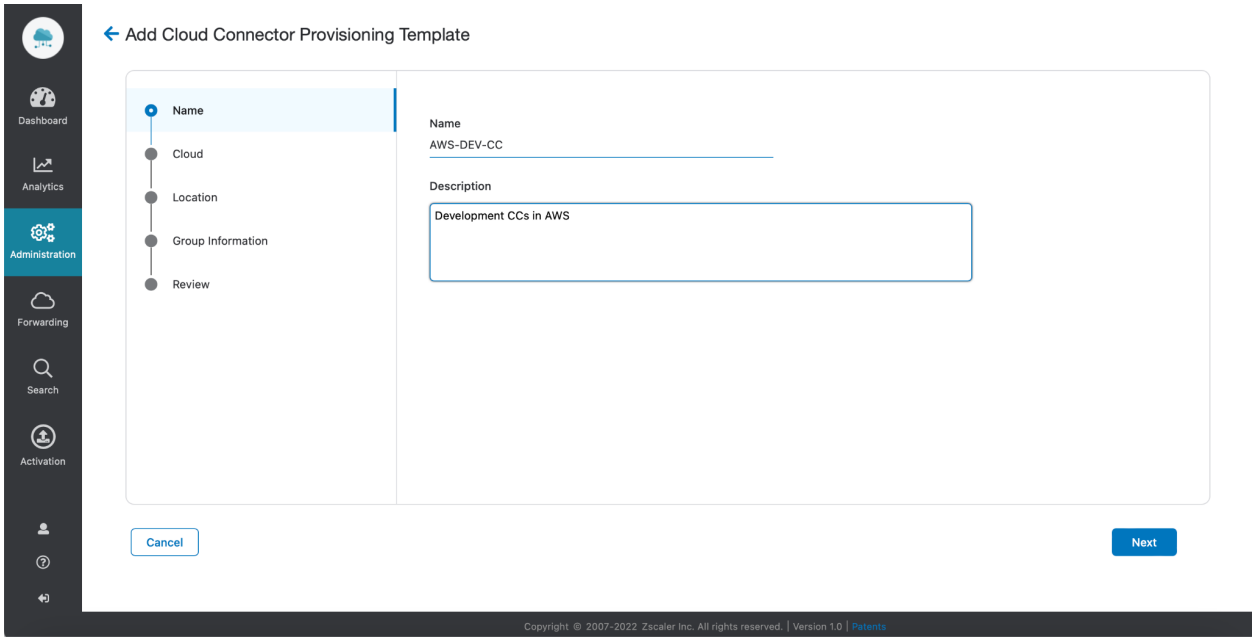
The glue that binds a cloud network to a Location Template and, hence, a Location and its configured attributes is a Provisioning Template. Navigate to Provisioning Templates under the

Administration menu. Here, you'll notice that an AWS template has already been configured for you. In fact, your currently registered Cloud Connectors used this template when registering.

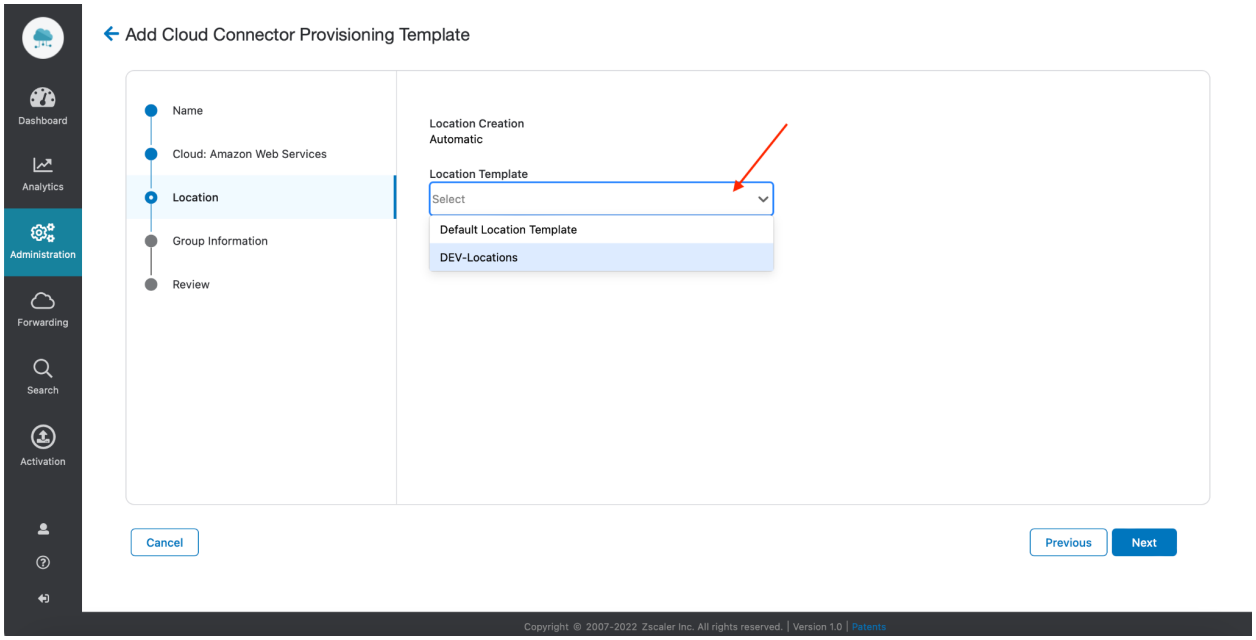
Click the Add Cloud Connector Provisioning Template button:



Provide a name and description and click the Next button. Choose AWS or Azure as the Cloud Service Provider and click the Next button (this template will not be used, so feel free to explore):

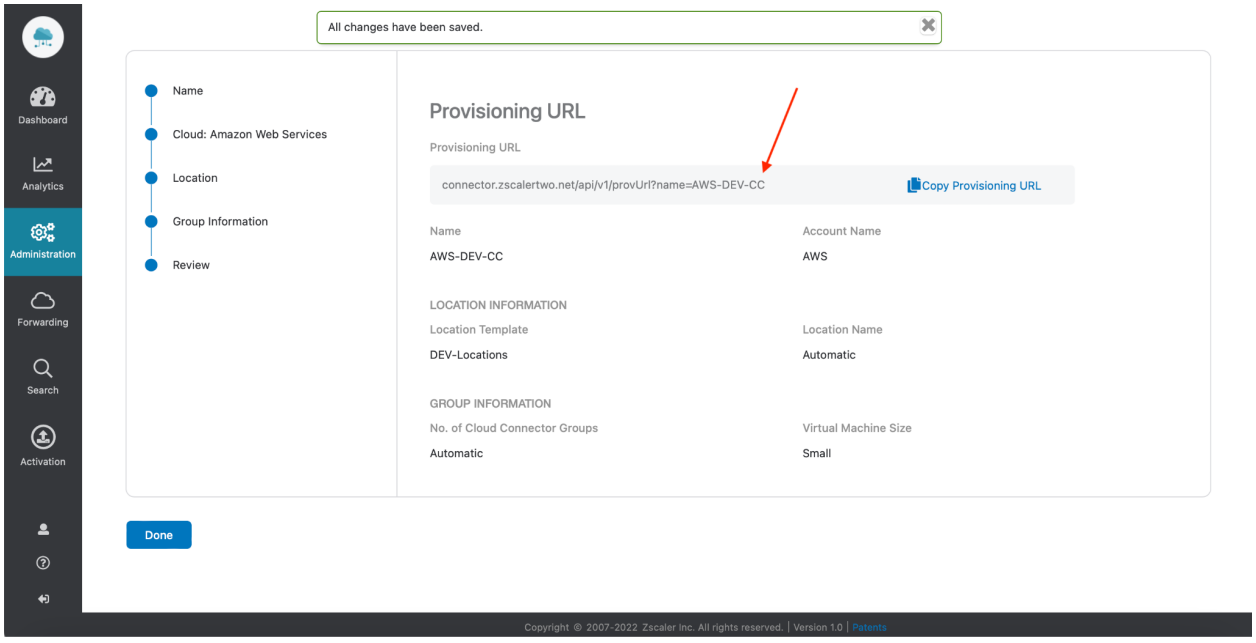


In the Location Template dropdown menu, select the Location Template you created in the previous steps and click the Next button:



Leave the VM size as Small, click the Next button again and complete the workflow by clicking the Save button.

In the screen that appears, note the Provisioning URL. This URL is used as part of the bootstrapping process of a Cloud Connector appliance. When the appliance boots, this URL tells the appliance how to “dial home” and inherit the correct Location settings:

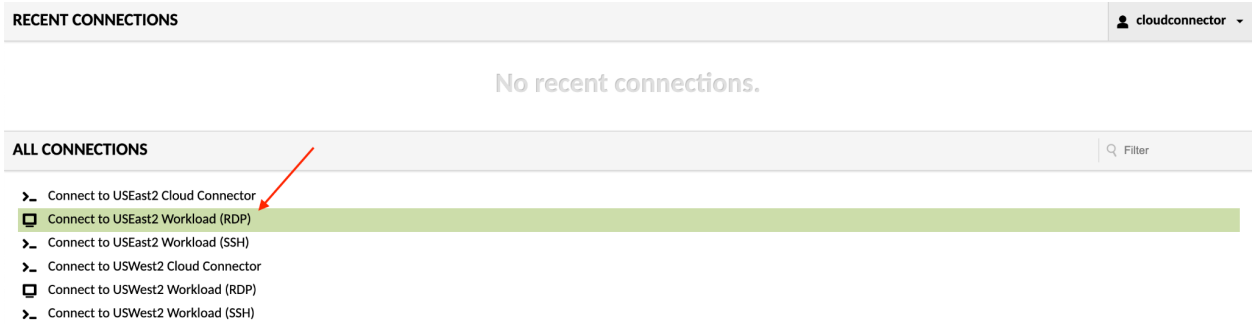


Traffic Forwarding Policy

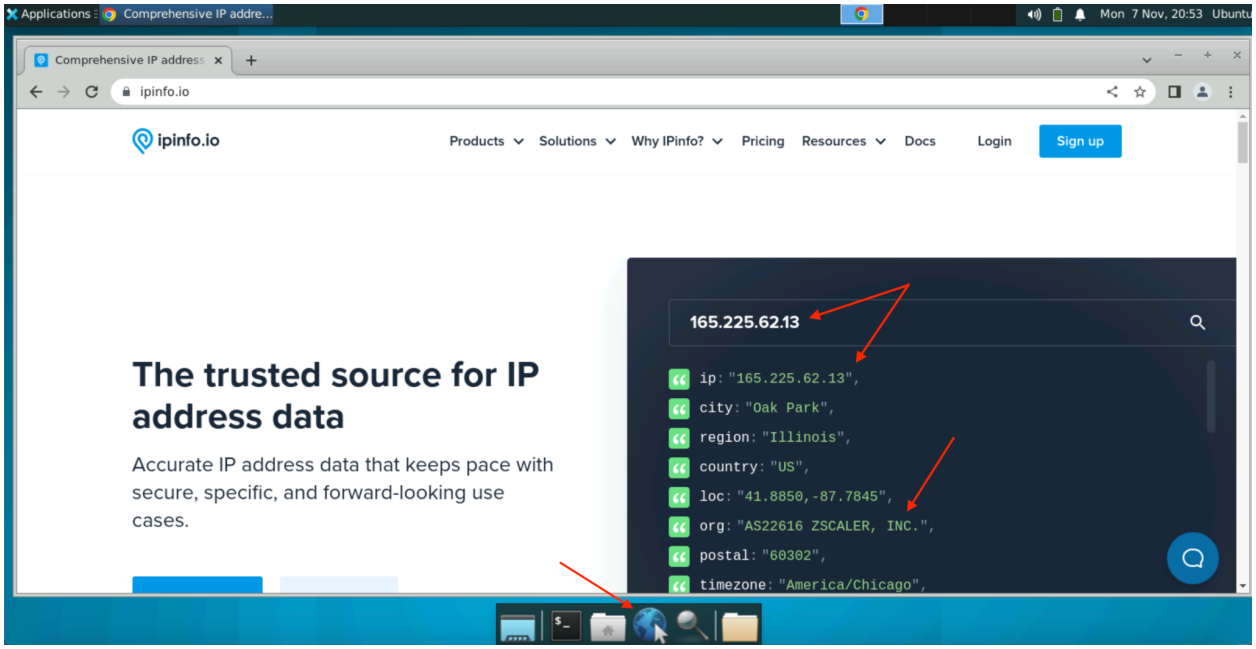
Throughout this lab, you’ll have access to two separate test machines: a Region1 Workload and a Region2 Workload. Both of these machines send their traffic to the internet via Cloud

Connector and, hence, the Zscaler Zero Trust Exchange, which includes ZIA. In this section, we'll experiment with Traffic Forwarding Policy within the Cloud Connector portal. Traffic Forwarding Policies allow you to manipulate how traffic is processed within the Cloud Connector.

From your Guacamole interface, access your Region2 Workload (RDP) by clicking on it:

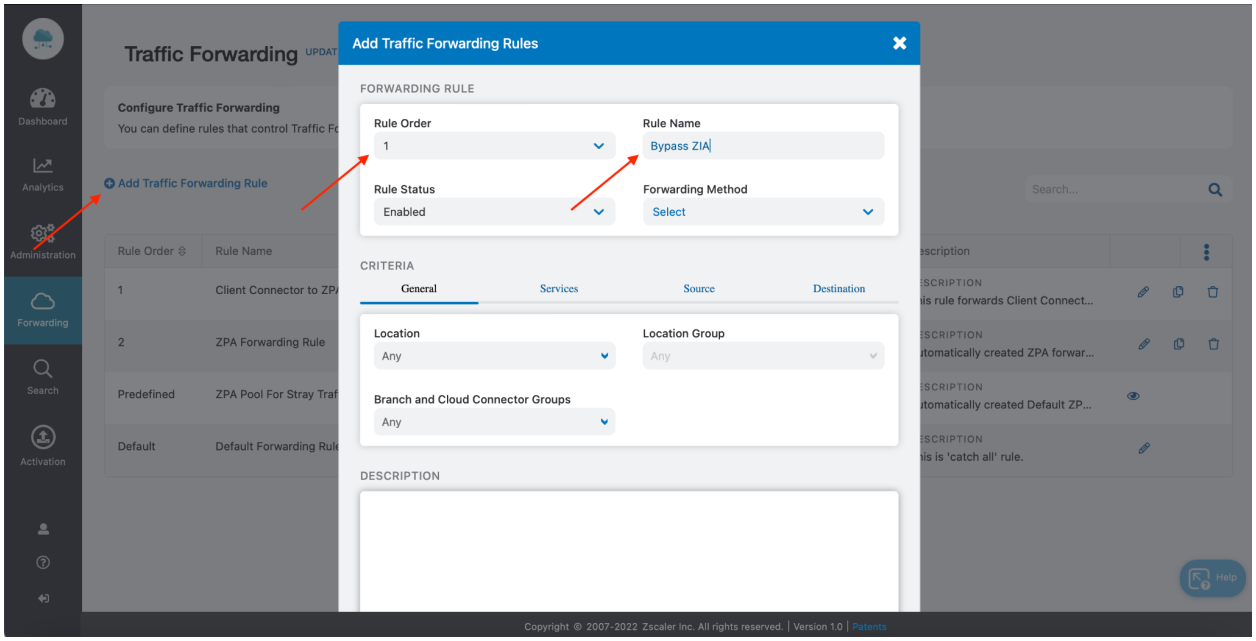


In the console of your machine, open up an internet browser and browse to https://www.ipinfo.io. Note the IP Address and Org of your workload. This traffic reached the internet via ZIA and used one of Zscaler's public IP addresses:

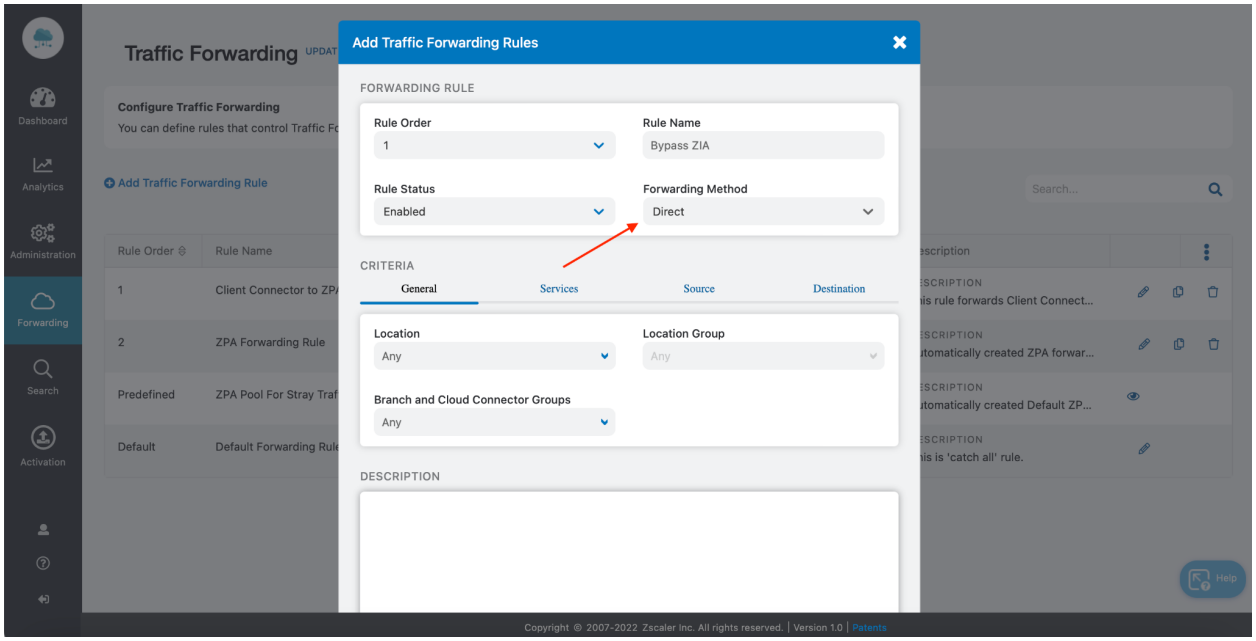


Let's assume that you'd like to prevent this traffic from using ZIA and, instead, go directly out to the internet from AWS. From your Cloud Connector portal find the Forwarding menu. Click on the Traffic Forwarding link. Here, you will notice a few default rules installed automatically when the Cloud Connector registered. These rules triggered when you sent traffic to https://www.ipinfo.io just now (specifically, the *Default* ZIA rule).

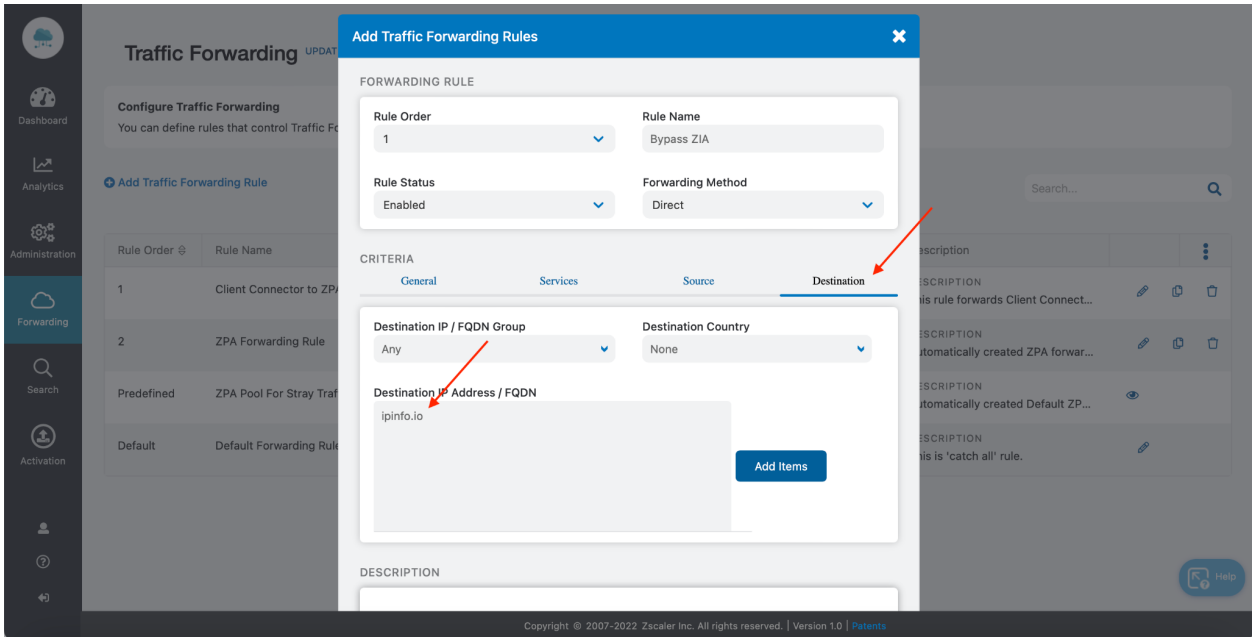
Click the Add Traffic Forwarding Rule button. Change the Rule order to 1 to ensure your new rule is placed at the top, and provide a Name:



In the Forwarding Method dropdown, choose the Direct option:



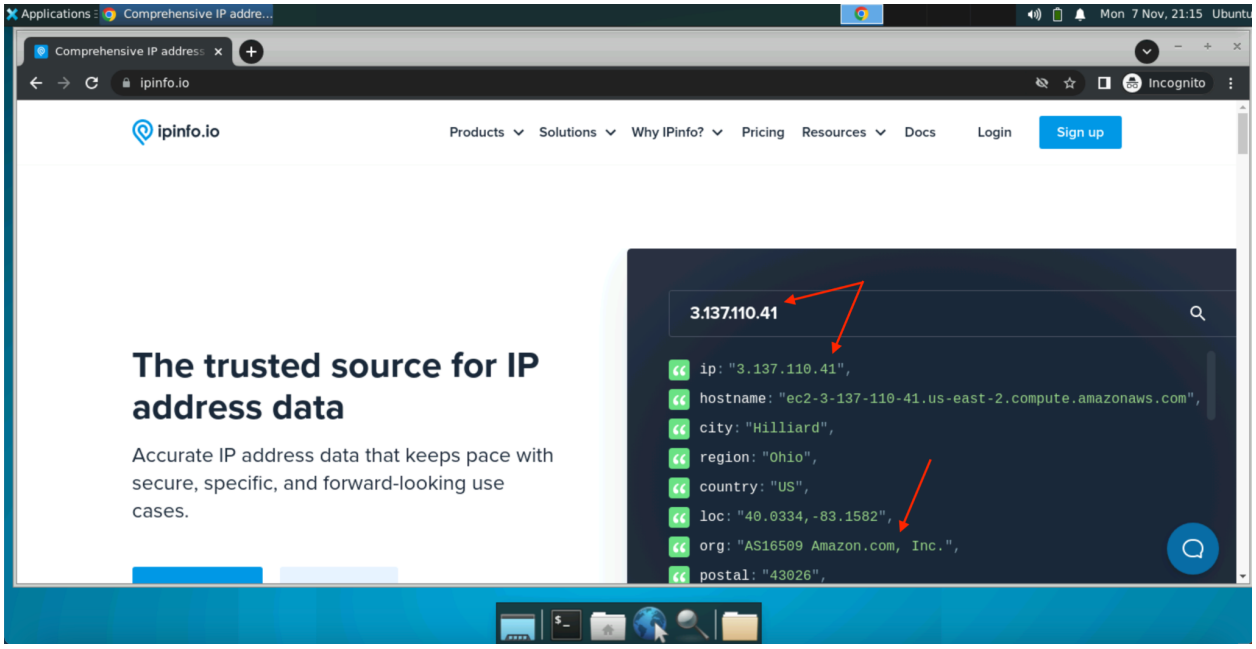
Notice, in the Criteria section, all of the options available for selecting which traffic will adhere to this rule. Click the Destination tab. In the Destination IP Address / FQDN field, enter ipinfo.io and click the Save button:



NOTE: You may have noticed other Forwarding Methods when you created the Direct rule. The Zscaler Internet Access (ZIA) option, as implied, will allow traffic matching the criteria defined to be forwarded to the ZIA cloud for inspection. By default, for ZIA customers, a rule will be automatically created for you to send all traffic to ZIA. The Zscaler Private Access (ZPA) option allows traffic that matches the criteria defined to be forwarded to the ZPA cloud. Cloud Connector automatically downloads ZPA Application Segments from your ZPA portal. Hence, any traffic it receives that is destined to these segments will be proxied, assuming it is permitted within the ZPA Access Policy and Client Forwarding Policy. Similar to ZIA, for ZPA customers, a default rule will be added automatically to ensure ZPA-bound traffic is automatically forwarded to the ZPA Broker.

NOTE: You may need to give the system 1-2 minutes for the change to take effect

Return to your USEast (Region2) Workload console and open up a new browser window (we recommend that this be an *Incognito* window to avoid browser caching issues). Navigate to https://www.ipinfo.io again and review the results. You should notice that this traffic now exits AWS directly as evidenced by the IP Address and Org. Feel free to experiment with additional Traffic Forwarding Rules:



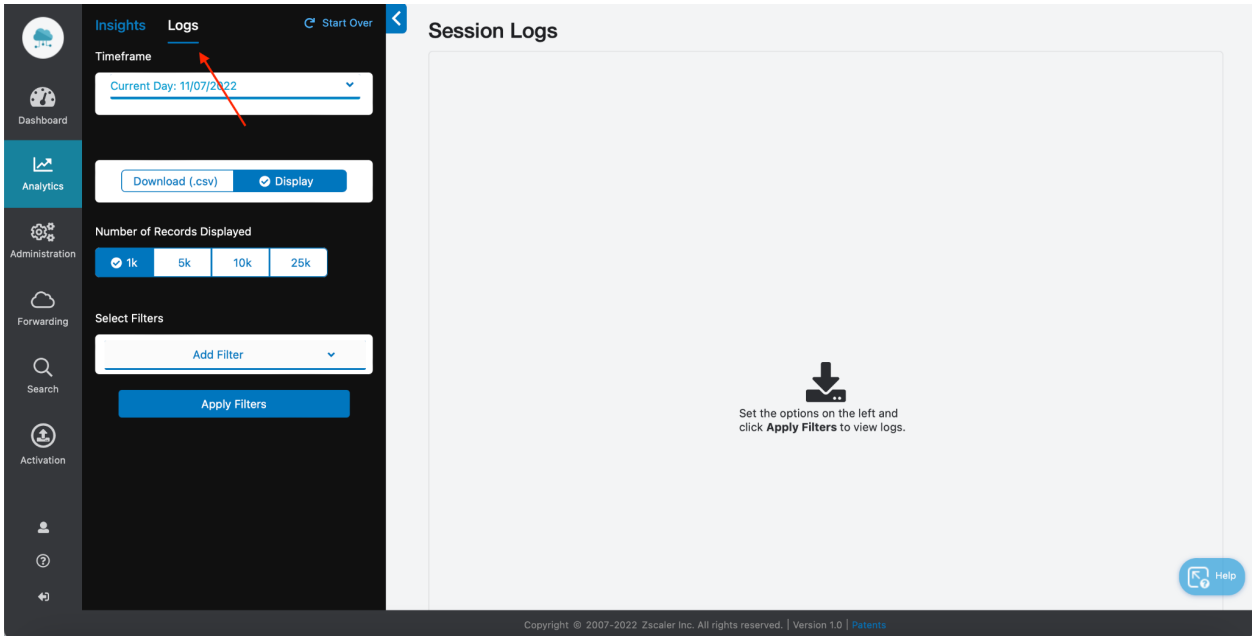
NOTE: You may have noticed other Forwarding Policy types in the Forwarding menu of the Cloud Connector portal. Log and Control Policies allow an administrator to identify control-plane traffic from specific cloud locations and redirect this traffic to a specified Zscaler Logging Gateway. DNS Policies find their usefulness with regards to ZPA use cases, which we'll discuss in a future lab. Cloud Connector proxies ZPA traffic via synthetic IP Addressing hosted within the appliance Administrators can use DNS Policies to allow, block, and forward DNS requests for ZPA-bound traffic. Furthermore, when forwarding to ZPA, DNS Policies also allow the administrator to specify the synthetic IP ranges used.

Whether using DNS Policies, Log and Control or Traffic Forwarding, each of the three options permits the administrator to define a range of match criteria. This lab focused only on Traffic Forwarding Policies, but the workflow remains roughly the same regardless of the policy chosen.

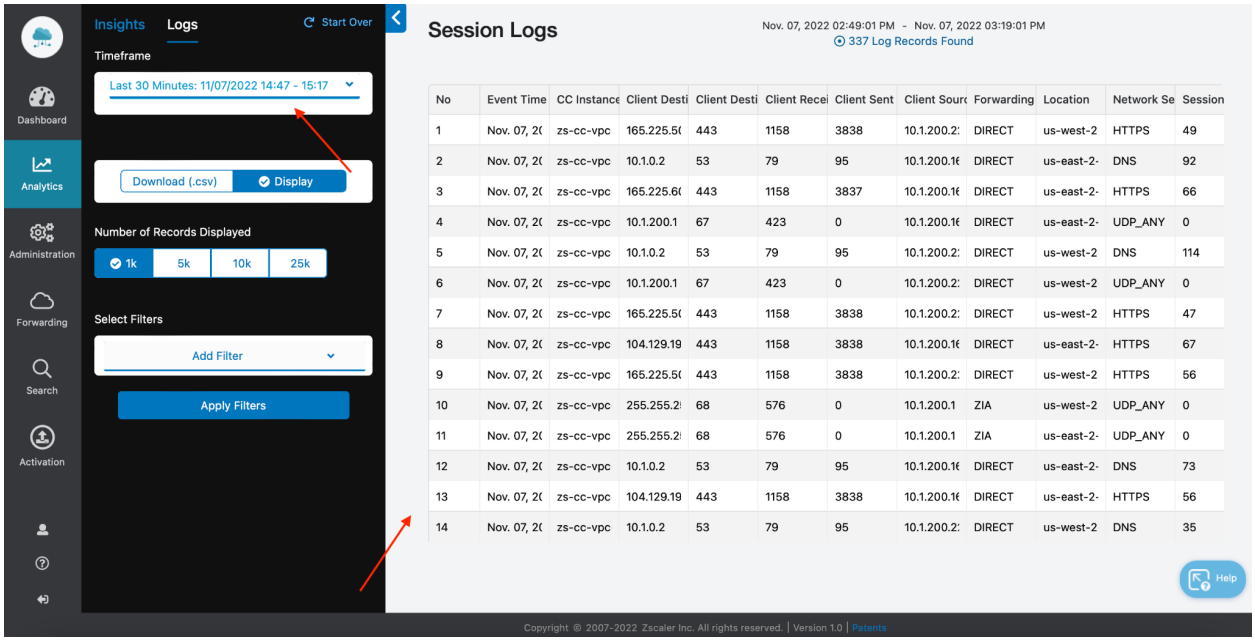
Analytics and Logging

Now that we've generated some traffic through our Cloud Connectors, let's review some of the logging that's available. Remember, Cloud Connector portal provides a new vantage point for logging of traffic before it reaches the Zero Trust Exchange. This allows you to ensure traffic not only reaches the appliance, but is routed to the correct Zscaler service.

Navigate to your Cloud Connector portal Analytics menu, followed by Session Insights. Click the Logs tab in the upper left:

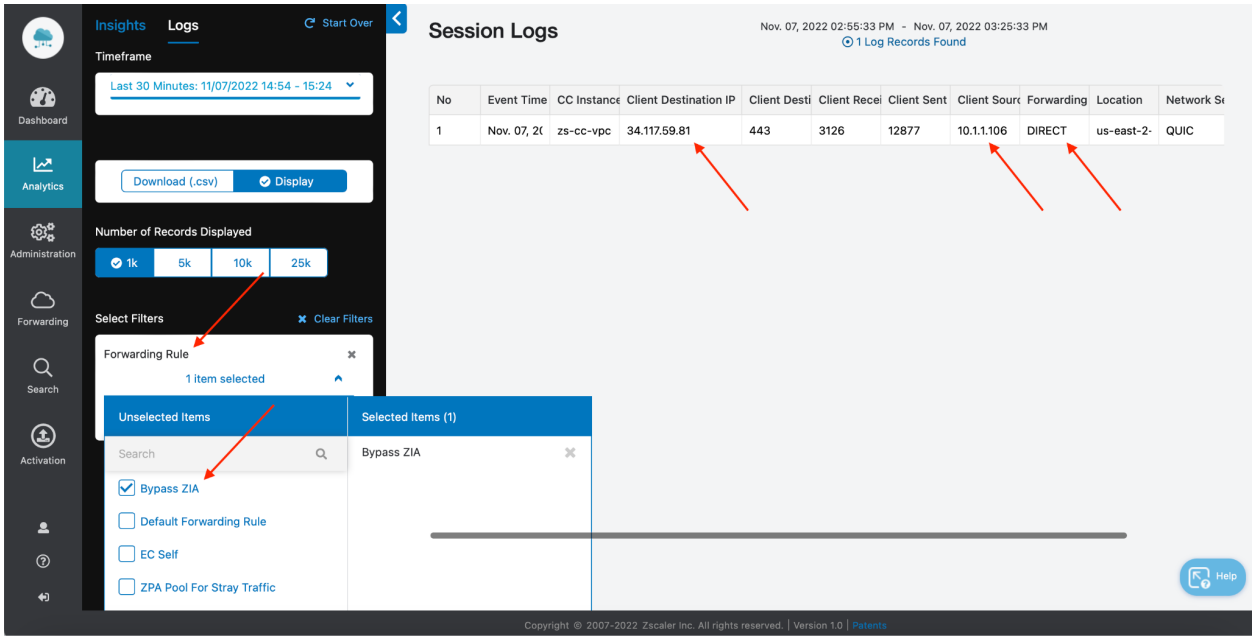


Change the Timeframe to the Last 30 Minutes and click the Apply Filters button:



Notice the logs that appear. Here, we can see 5-tuple information on flows seen by the Cloud Connector as well as the disposition of the traffic (where it was sent).

Feel free to experiment with the filtering options available. Can you find the traffic to https://www.ipinfo.io you sent? (Hint, try filtering by Forwarding Rule…)



NOTE: DNS Insights provides visibility into DNS traffic that crosses the appliance. This is particularly useful in ZPA use cases in which the appliance is proxying traffic using synthetic IP addresses, but it also provides a bit of visibility into the domains being queried by cloud workloads that are outside the organization. You'll find information on the DNS request itself, the resolved IP, and the disposition of the traffic. Tunnel Insights provides a glimpse into the data tunnels that are created from the appliance toward the Zero Trust Exchange. Here, you can view the source VPC or VNet the Cloud Connector sits within, its public IP, and the Zscaler IP address used to terminate the far end of the data tunnel.

Module 2: Protecting Cloud Workloads with ZIA

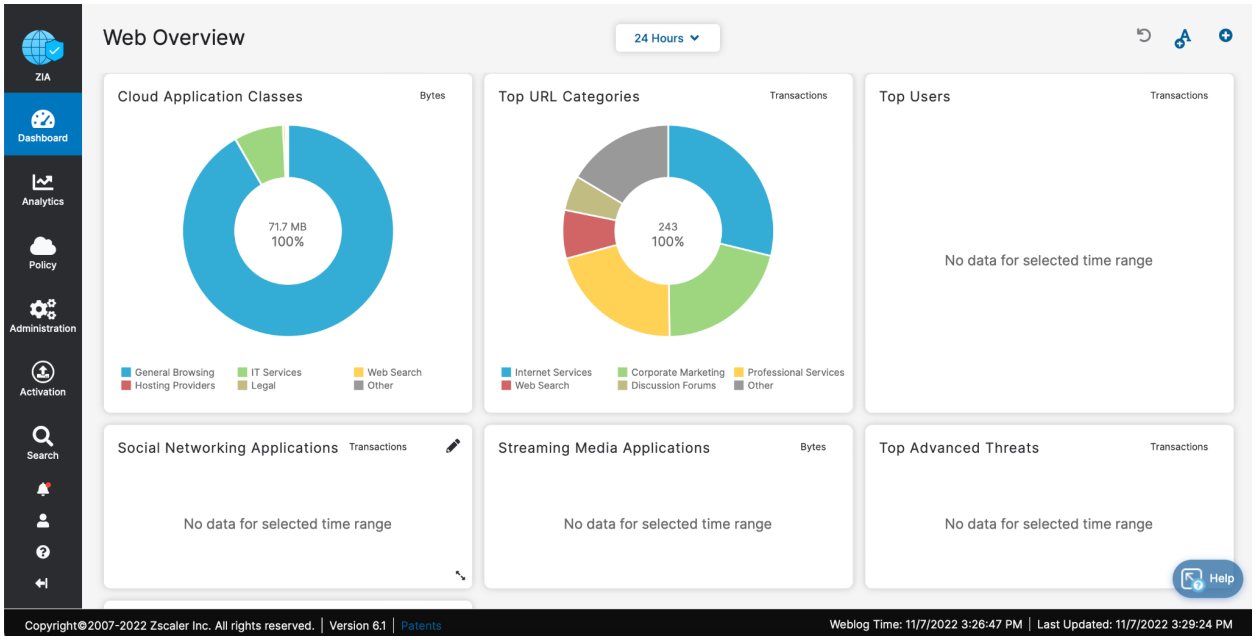
Now that we have traffic flowing through the Cloud Connector and into ZIA, let's create some security policy to keep our cloud workloads secure.

Enforcing Minimum TLS Versions

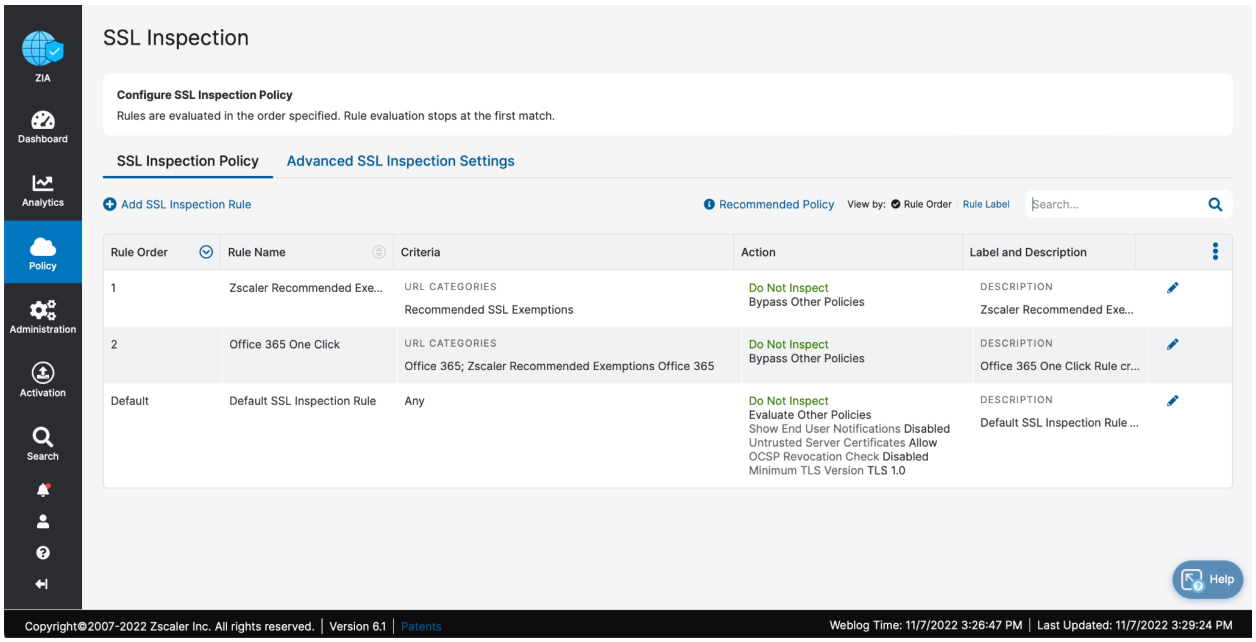
With known vulnerabilities in earlier versions of TLS—namely TLS 1.0 and 1.1—these protocol versions should no longer be used. Let's configure a policy to enforce the minimum TLS version as 1.2.

Navigate to the ZIA portal and log in using the same credentials used for the Cloud Connector portal:

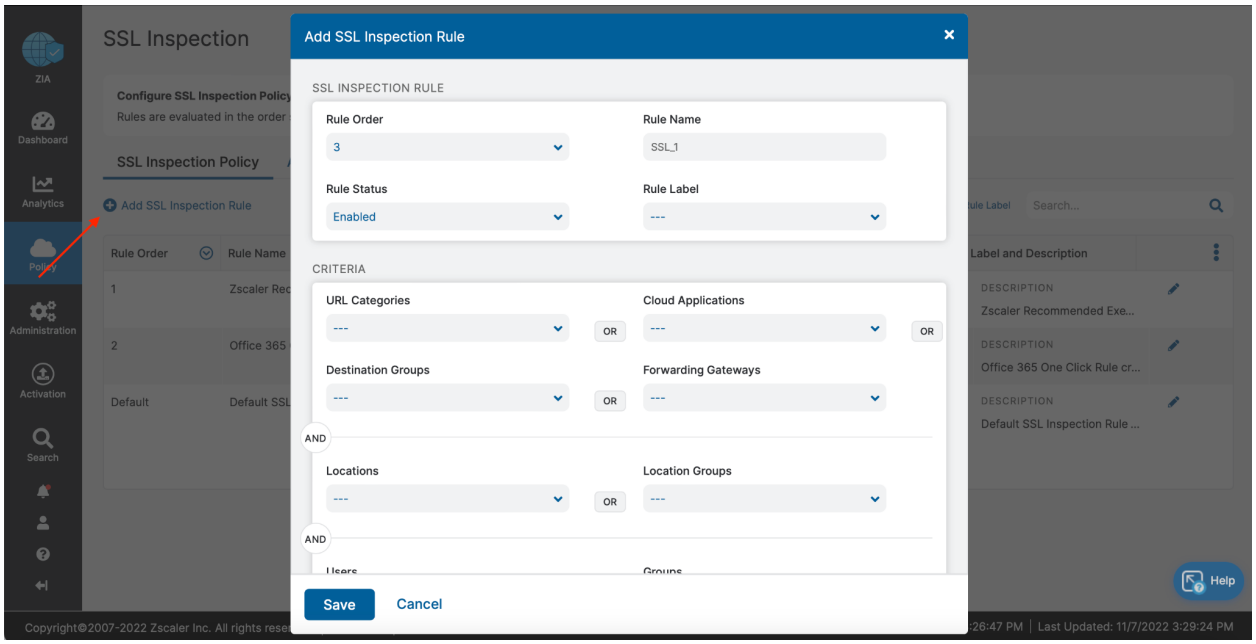
Zscaler Internet Access (ZIA) portal **Username**: student@relacszXX.net
Zscaler Internet Access (ZIA) portal **Password**: 7f#1CQ^DU03K



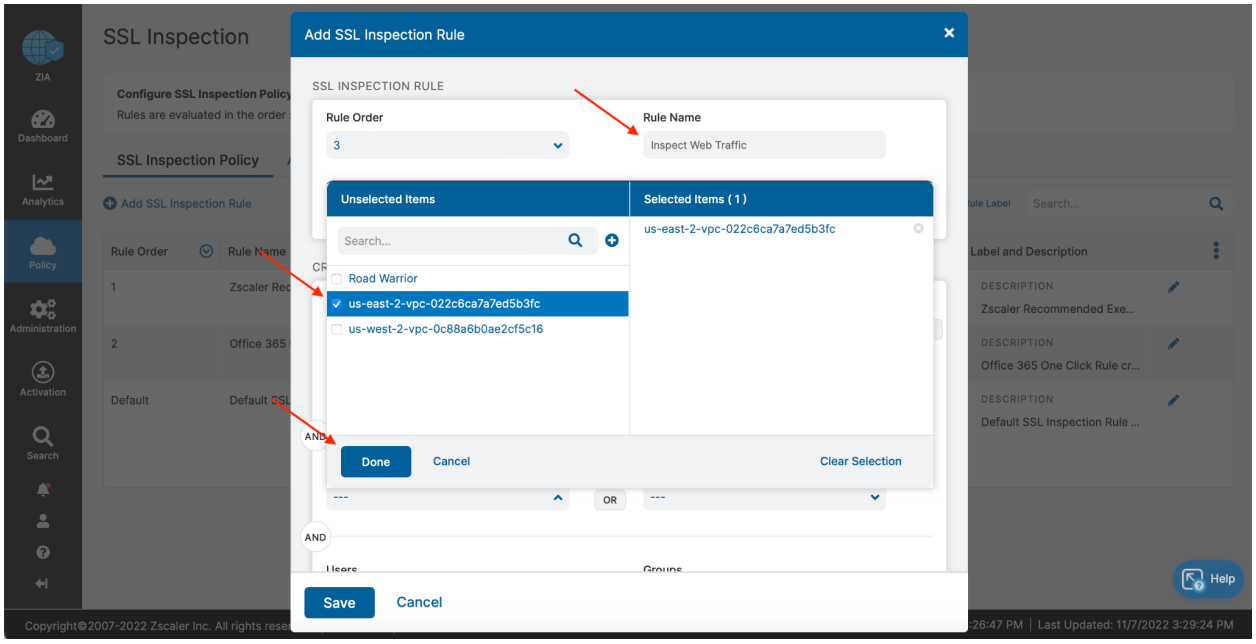
Navigate to the Policy menu, followed by SSL Inspection:



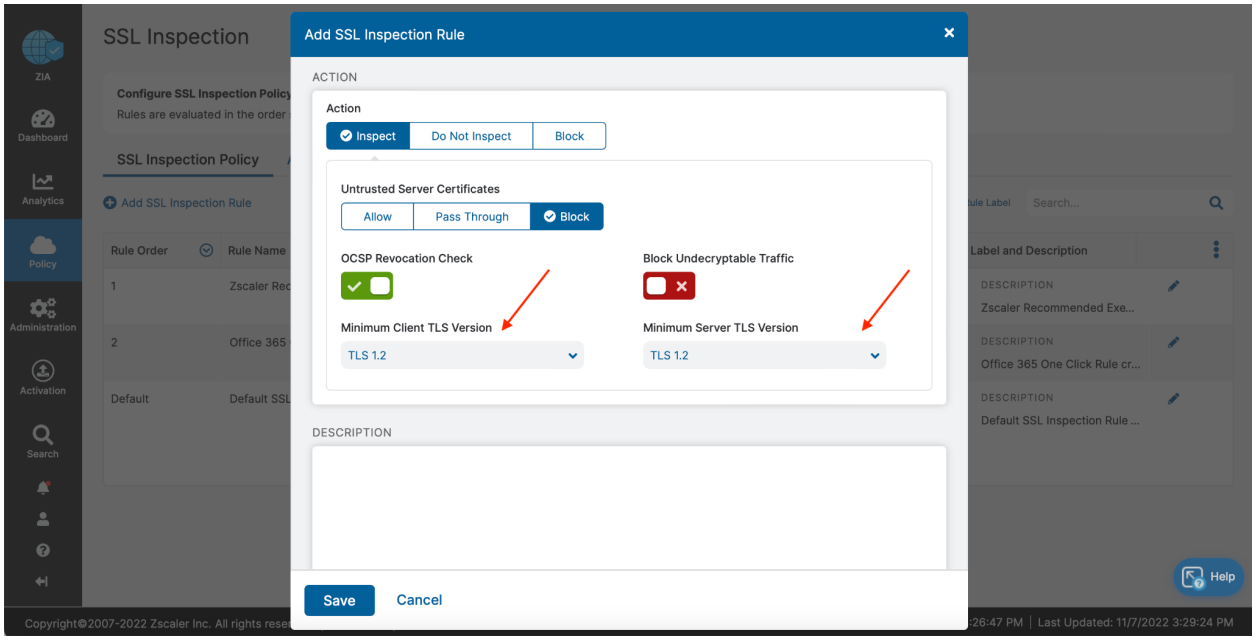
As you can see from the rules listed, the default behavior is to not inspect SSL traffic. However, you will now create a rule to change that. Click on Add SSL Inspection Rule:



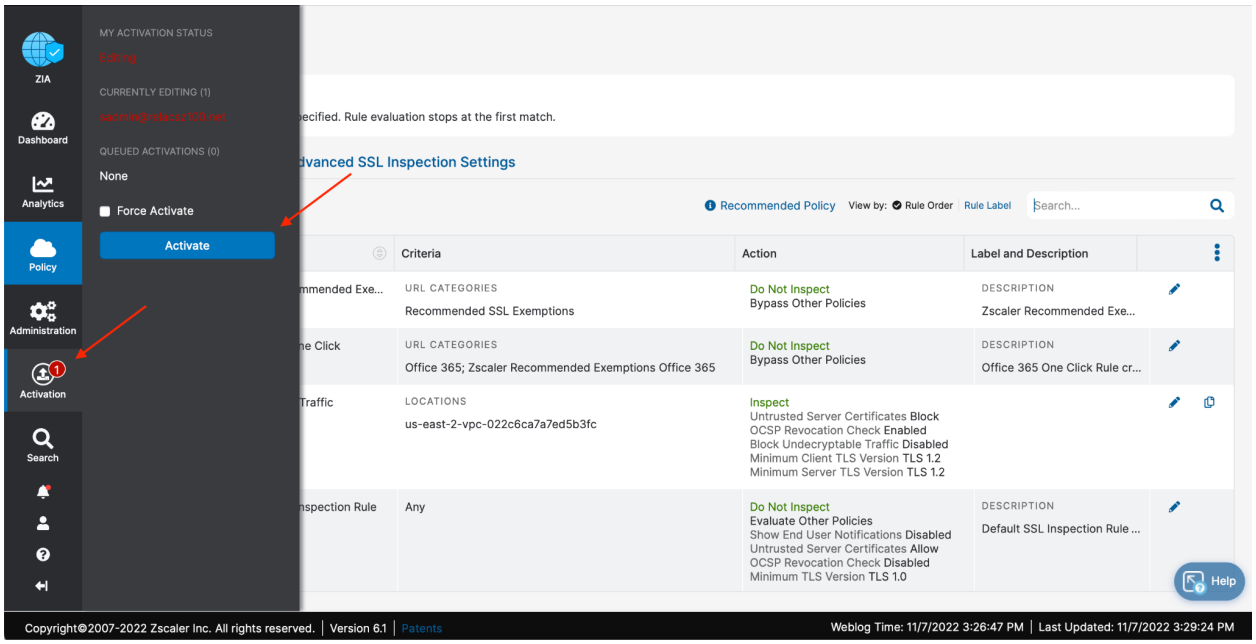
Provide a name. Under the Criteria section, click on Locations and select your US-East Location. The Locations listed take on the naming convention of Region + VPC ID. Selecting US-East ensures that this policy only applies to your US-East Workload. You also have the option of selecting multiple Locations, or not selecting a Location at all. In this case, this policy would be applied to multiple, or all Locations, respectively. Click the Done button, when finished:



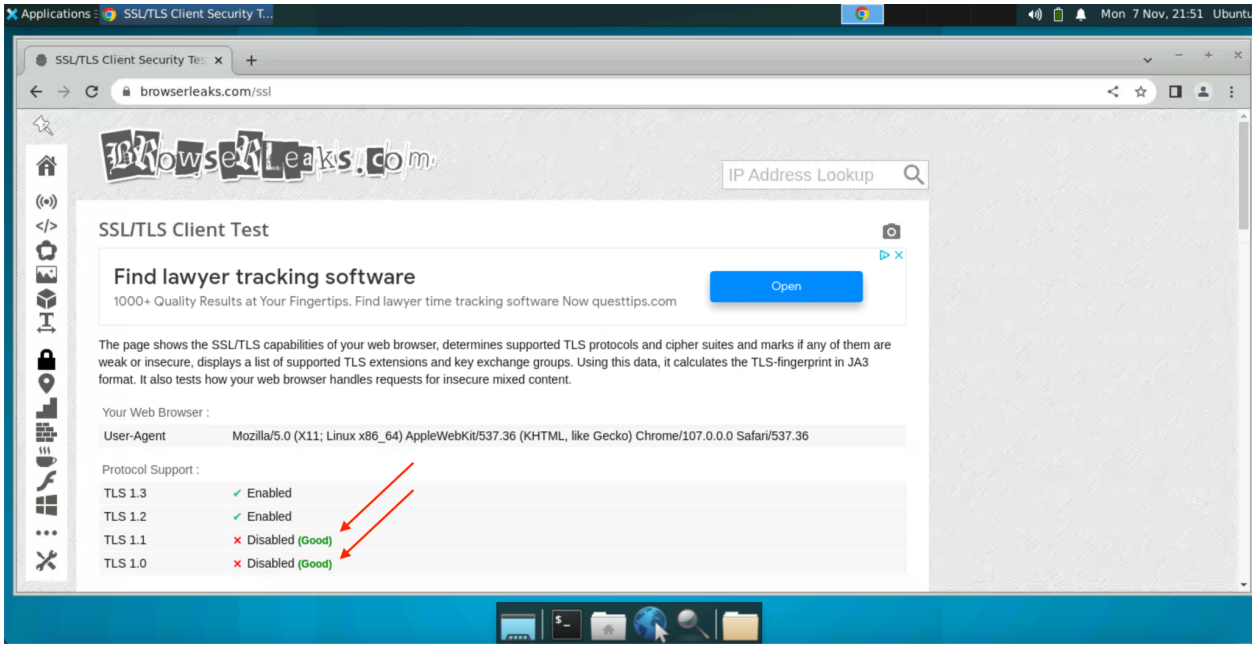
Scroll down to the Action section and, under Minimum Client TLS Version and Minimum Server TLS Version, select TLS 1.2:



Click the Save button and activate your changes by hovering over the Activation menu and clicking the Activate button:

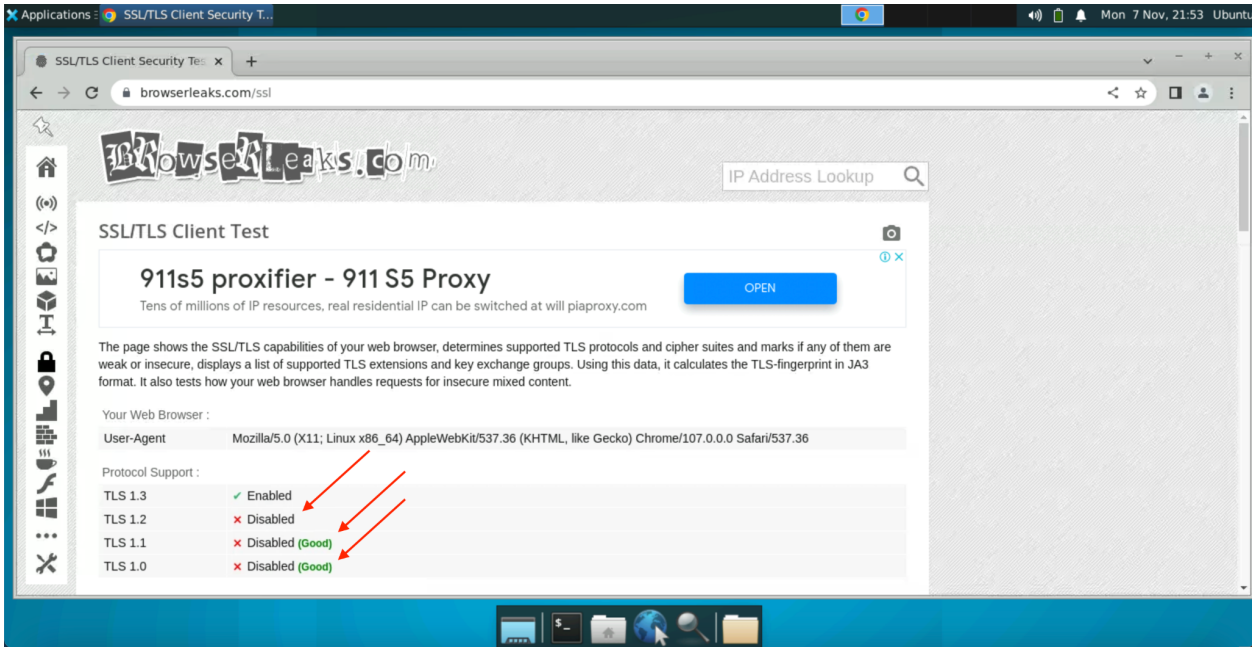


Next, return to the Region2 (USEast) Workload, open an internet browser and navigate to https://www.browserleaks.com/ssl. The site will automatically begin to test the TLS versions enabled and accessible on your browser. Note that TLS v1.2 and TLS v1.3 are enabled. TLS v1.0 and TLS v1.1 are disabled, as expected:

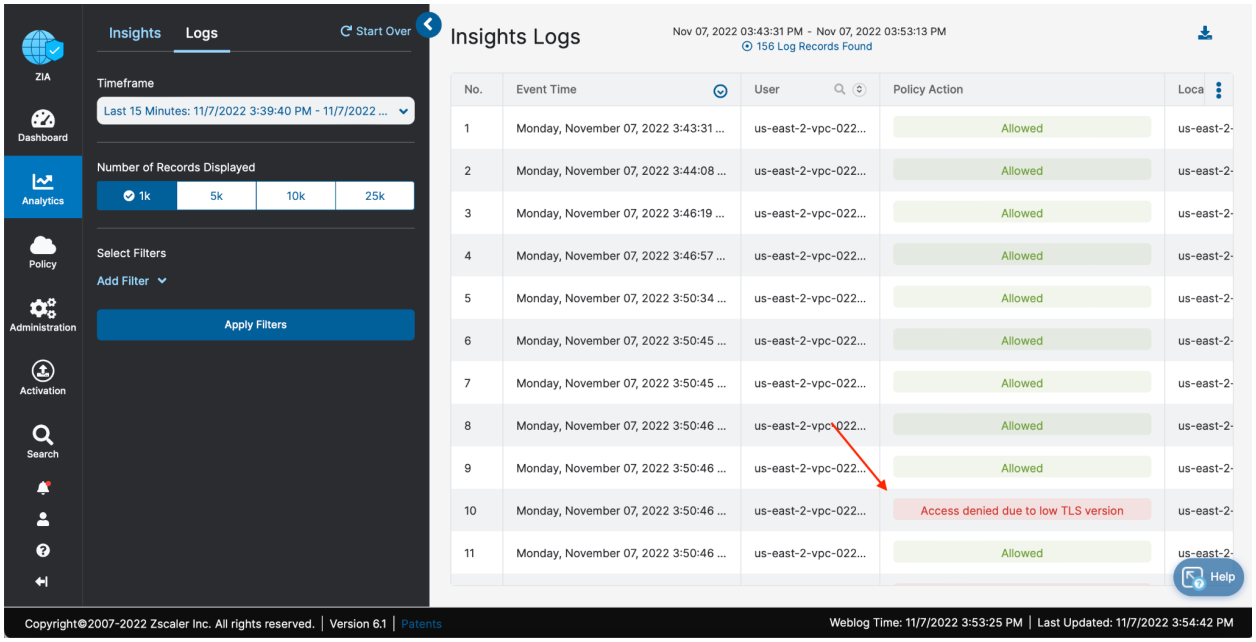


Return to the ZIA portal, SSL Inspection screen, via the Policy menu. Click the “pencil” icon next to your SSL policy to edit it. Scroll to the Action section and, under Minimum Client TLS Version and Minimum Server TLS Version, select TLS 1.3.

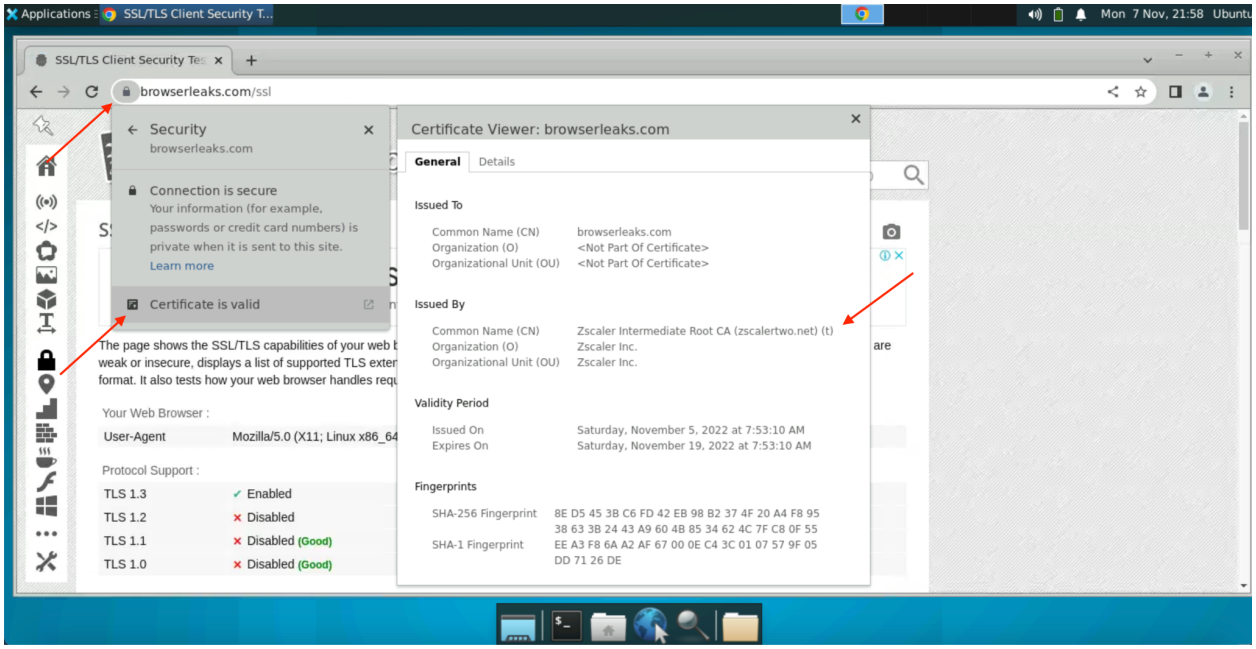
Return to the USEast Workload and repeat the test to https://www.browserleaks.com/ssl. You should notice that TLS v1.0, v1.1, and v1.2 are now disabled, leaving only TLS v1.3:



You can verify this behavior from the ZIA portal via the Analytics menu under Web Insights. In the Logs tab (upper left), narrow the Timeframe to the last 15 minutes and click the Apply Filters button:



Zscaler proxies SSL connections by inserting itself into the HTTPS exchange. Return to the USEast Workload console and click the “lock” icon in the upper left (next to the address bar). Click the ‘Connection is secure’ link, followed by ‘Certificate is valid.’ Notice that the certificate presented to the workload is actually that of Zscaler, rather than that of the requested website:

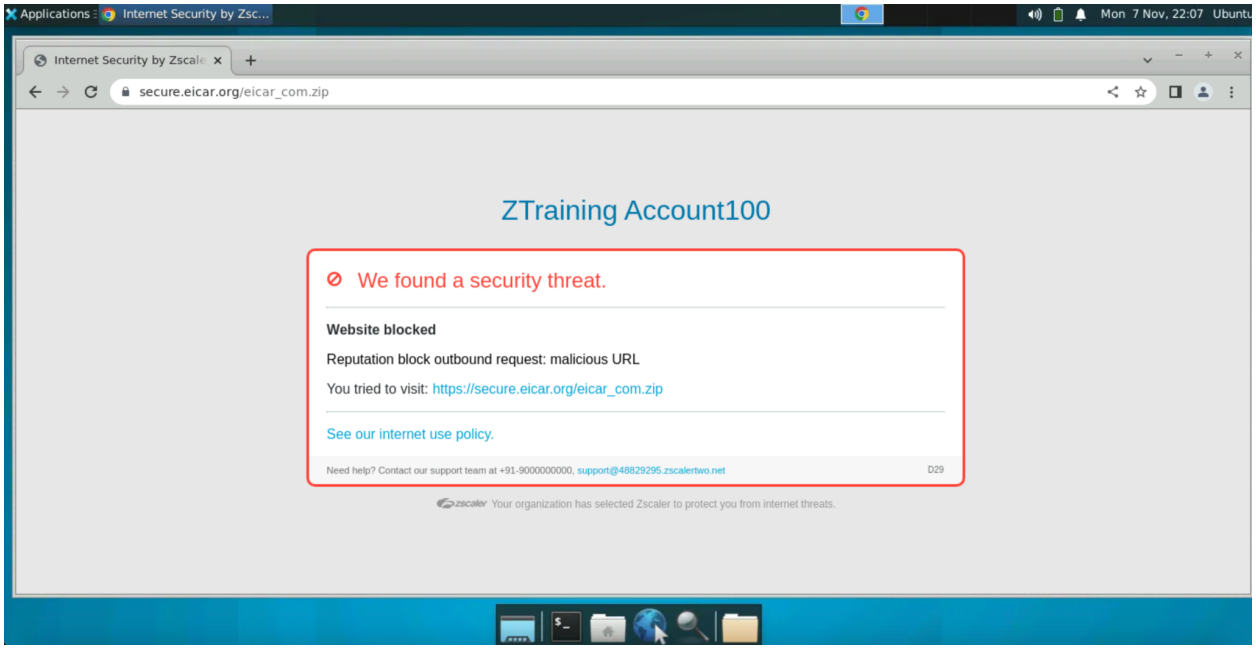


The Zscaler Zero Trust Exchange terminated the connection between your workload and itself, then created a new connection out to the requested website. This allows ZIA to perform man-in-the-middle inspection of encrypted traffic.

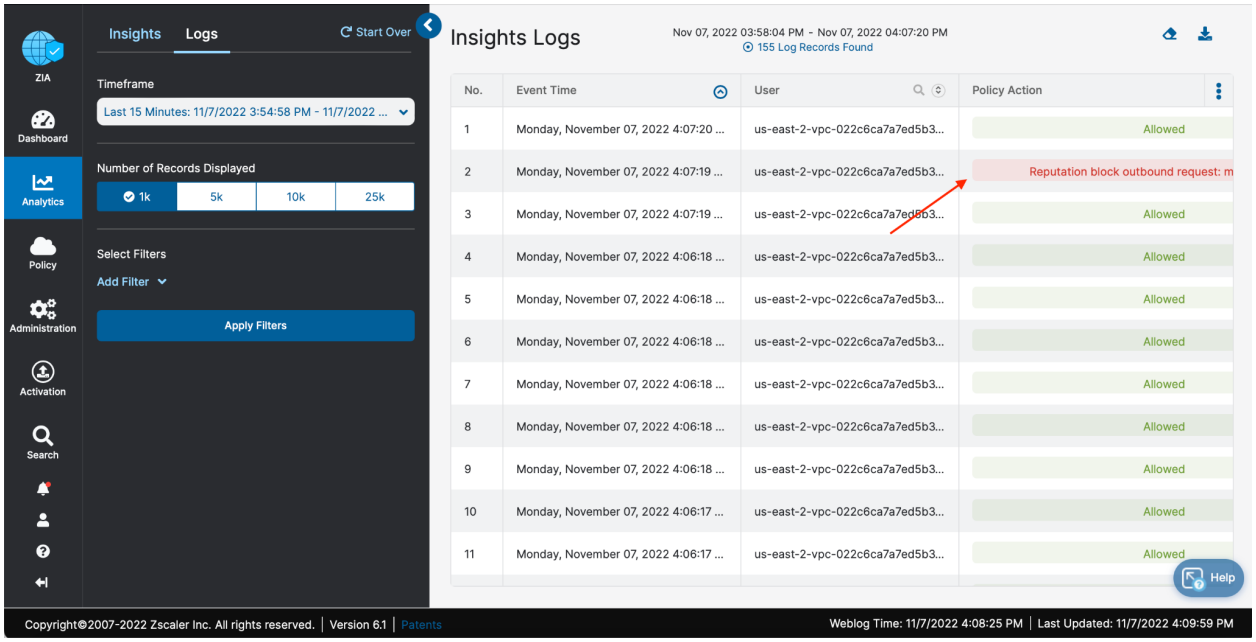
Protecting Against Malicious Payloads, Phishing, and BotNet

So far, you've explored SSL Decryption and enforced minimum TLS versions. Let's now look at what protection Zscaler offers out of the box.

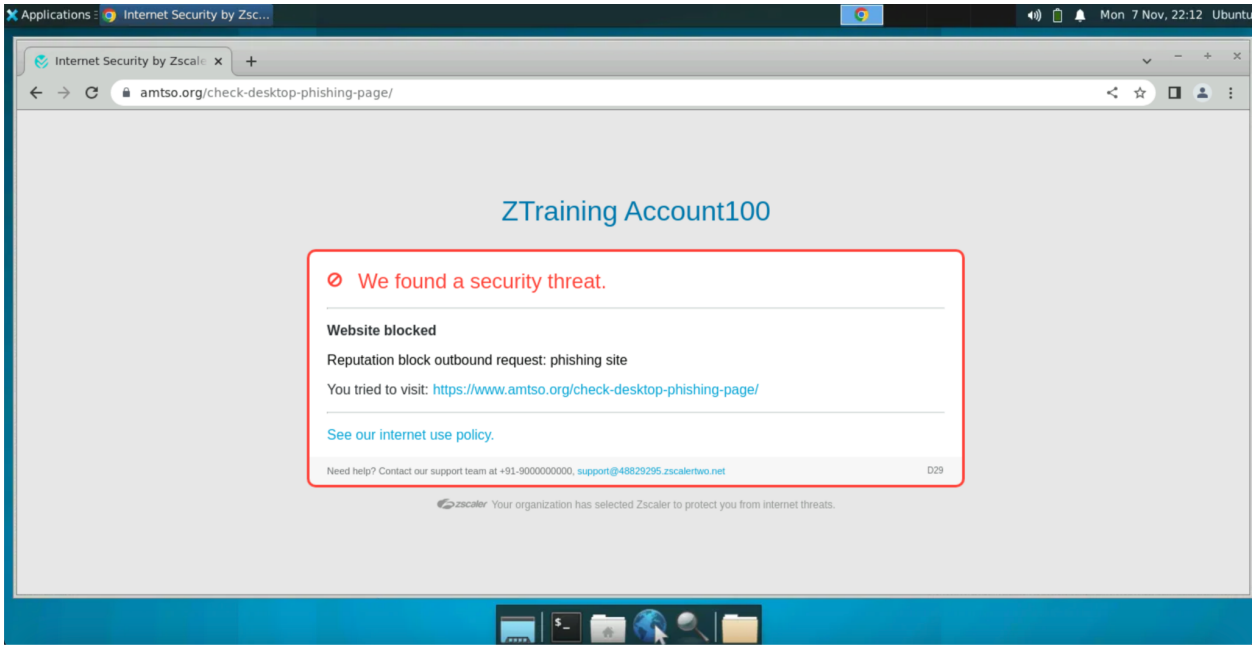
For this test, you will use a well-known test file from EICAR (European Institute for Computer Anti-Virus Research). From the USEast (Region2) Workload, browse to https://www.eicar.org/download-anti-malware-testfile. On the right side of the page, click one of the test files to download (eicar.com, eicar.com.txt, eicar_com.zip, etc.). You should notice that Zscaler Internet Access automatically blocks this download attempt:



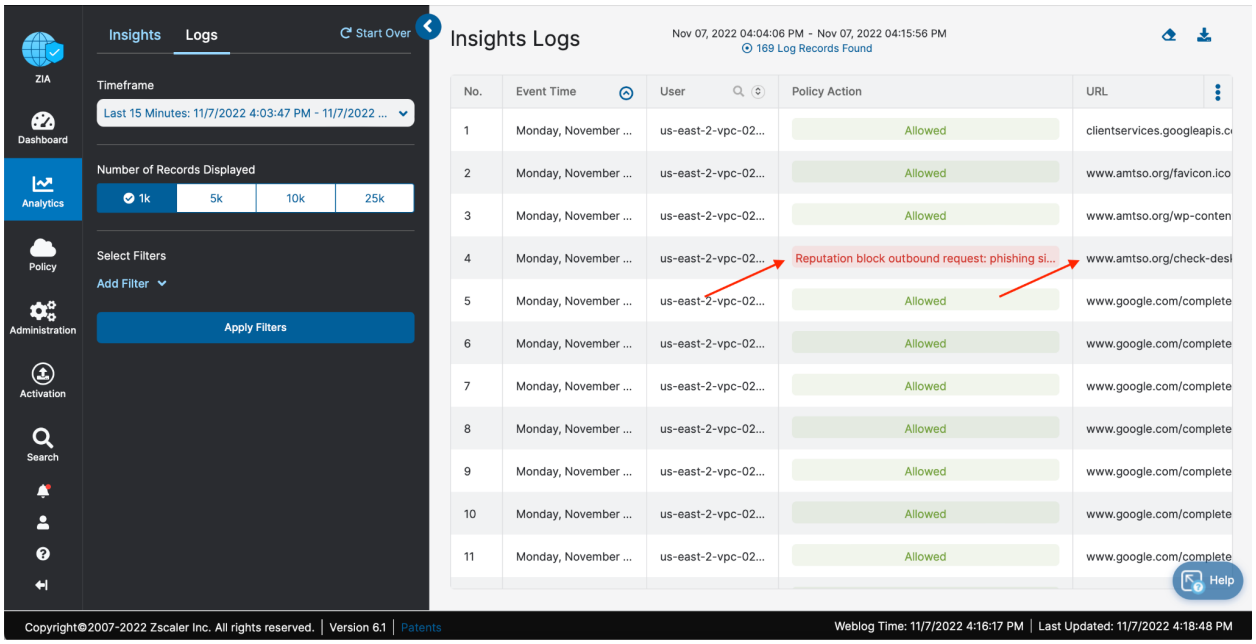
This is because the file contained malicious code. Even malicious files hidden inside compressed archives (ZIP) can be blocked. Again, from the ZIA portal Analytics (Logs tab) page, you can see the log entry for the file you just attempted to download (be patient, it may take 2-3 minutes for logging to catch up):



Apart from viruses, Zscaler can also detect other threats such as phishing sites and BotNets. From your USEast (Region2) Workload, navigate to the following website: https://www.amtso.org/check-desktop-phishing-page. Note that the threat was blocked since this is a known phishing site:



Again, from the ZIA portal Analytics (Log tab) page, you can see the log entry for the site you just attempted to visit (be patient, it may take 2-3 minutes for logging to catch up):



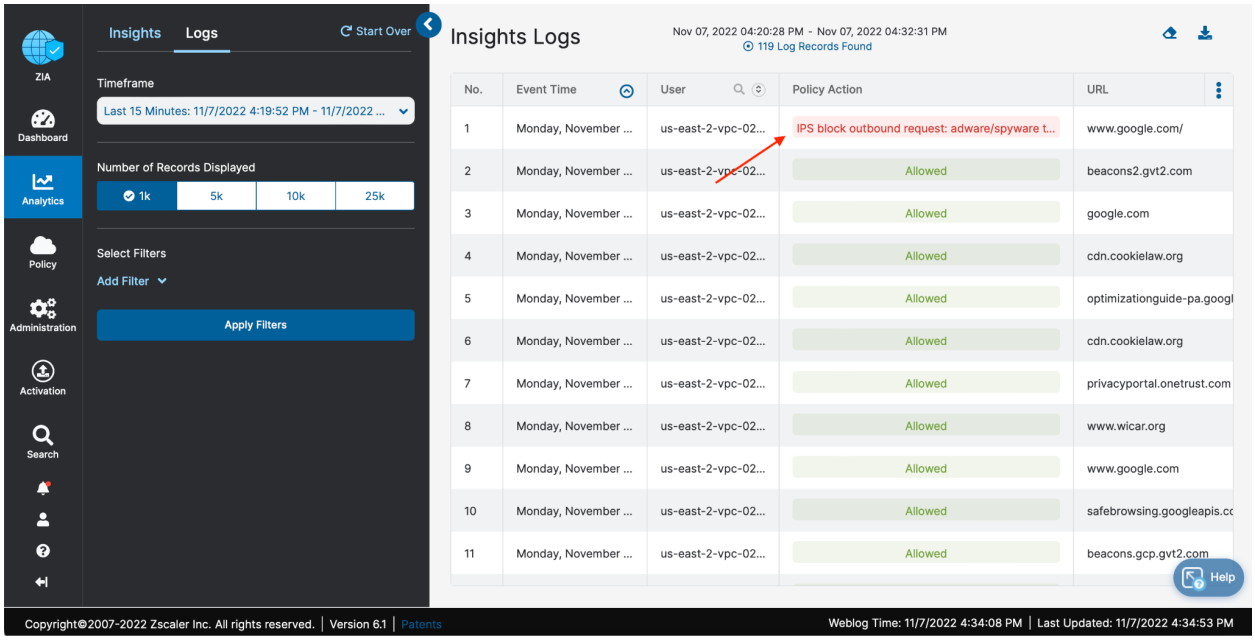
Lastly, return to the Guacamole interface and navigate to the Region2 Workload (SSH) connection. Enter the following command at the command line:

curl -A "BlackSun" www.google.com

This is a test that will trigger an IPS response from most firewalls. This too should be blocked by ZIA IPS automatically, as shown below:

```
<!--locale en_US-->
<table id="en_US" width="100%" border="0" cellspacing="0" cellpadding="0">
<tbody><tr><td class="eu_h">
<i class="a_i"></i>
We found a security threat.
</td></tr>
<tr><td class="hr"><hr></td></tr>
<tr><td class="eu_co">
<b>Website blocked</b>
</td></tr>
<tr><td class="eu_co rsn">
IPS block outbound request: adware/spyware traffic
</td></tr>
<tr><td class="eu_co">
You tried to visit:<div class="eu_l"><a href="http://www.google.com/">http://www.google.com/</a></div>
</td></tr><tr>
<td class="hr"><hr></td>
</tr>
<tr><td class="eu_co ln">
<a href="http://48829295.zscalertwo.net/policy.html">
See our internet use policy.
</a>
</td></tr>
<tr><td class="eu_co fo">
Need help? Contact our support team at +91-9000000000, <a href="mailto:support@48829295.zscalertwo.net">support@48829295.zscalertwo.net</a>
</td></tr>
<tr><td class="eu_co st red">
<span class="s_img"></span>
Your organization has selected Zscaler to protect you from internet threats.
</td></tr>
</tbody></table>
<!--/locale en_US-->
</td></tr>
</tbody></table>
</div>
</div>
</body></html>
<!-- 0 0 64 0 1667860352 4 http://www.google.com/ -->cloudconnector@ip-10-1-1-106:~$
```

From the ZIA portal Analytics (Log tab) page, you can see the log entry for the site you just attempted to visit (be patient, it may take 2-3 minutes for logging to catch up):

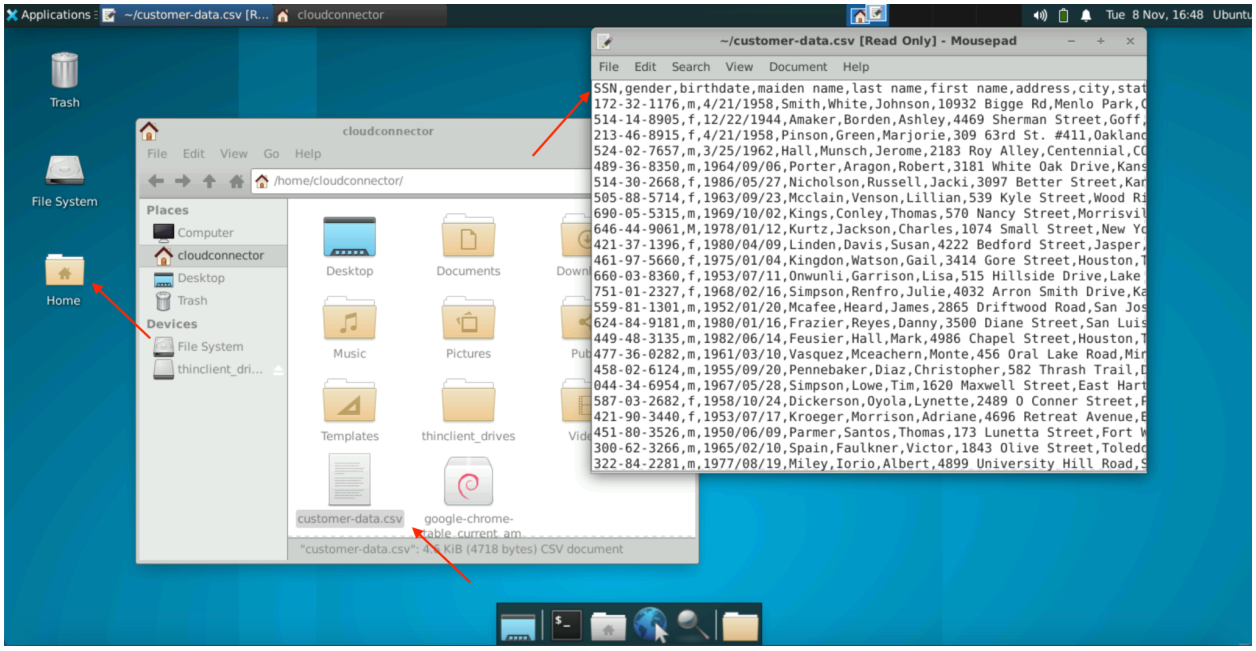


Though these are simple tests, the above exercises help demonstrate what Zscaler can do straight out of the box without requiring additional configuration. Aside from enabling SSL decryption, Zscaler automatically identifies and blocks threats without requiring administrators to build a complex policy first.

Enforcing a Data Loss Prevention Policy

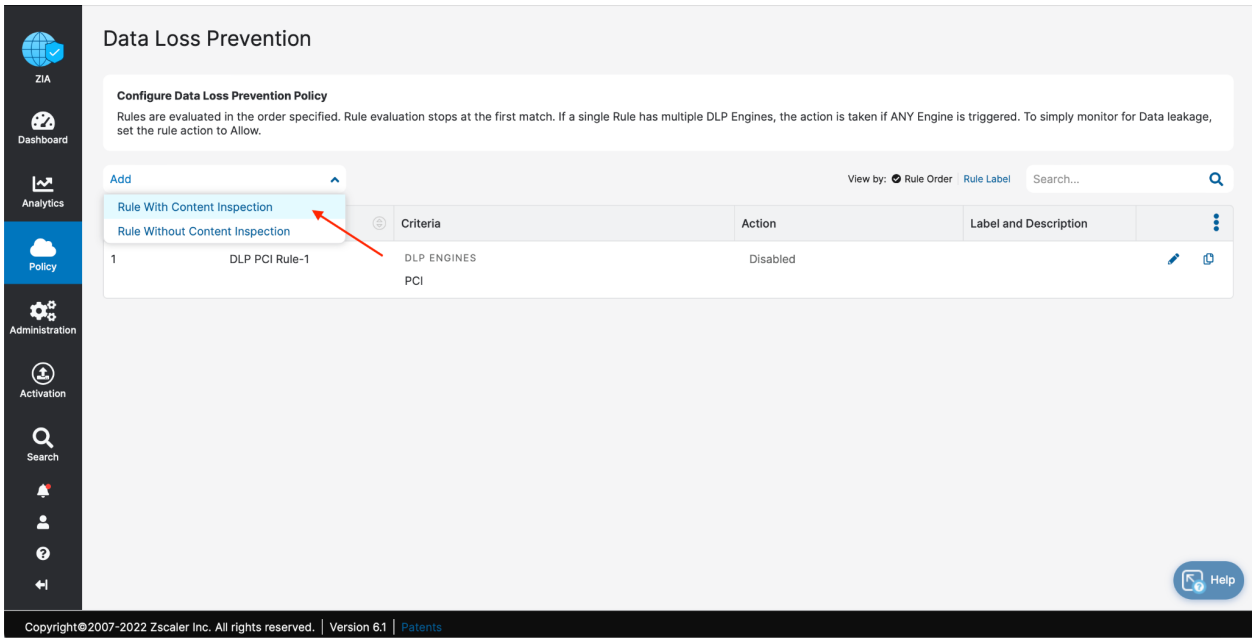
We've demonstrated how Zscaler protects your workload from external threats such as viruses, phishing sites, and BotNets. However, what if the bad actor originates from within your organization, or perhaps a workload has been compromised? In this section, we demonstrate how Zscaler prevents data exfiltration.

On the USEast (Region2) Workload, take a look at the customer-data.csv file located in the Home folder on the Desktop by double-clicking it (ensure you review the CSV file and not the TXT file):

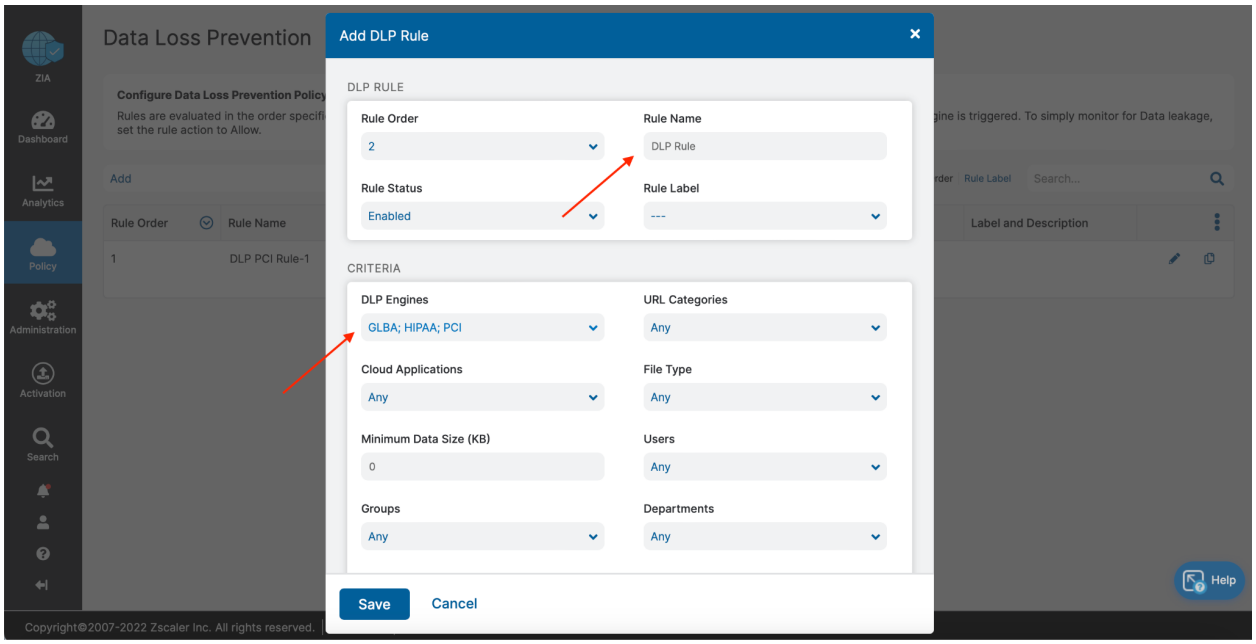


This file contains fake Personal Identifiable Information (PII). Let's assume, a disgruntled employee tries to exfiltrate this data.

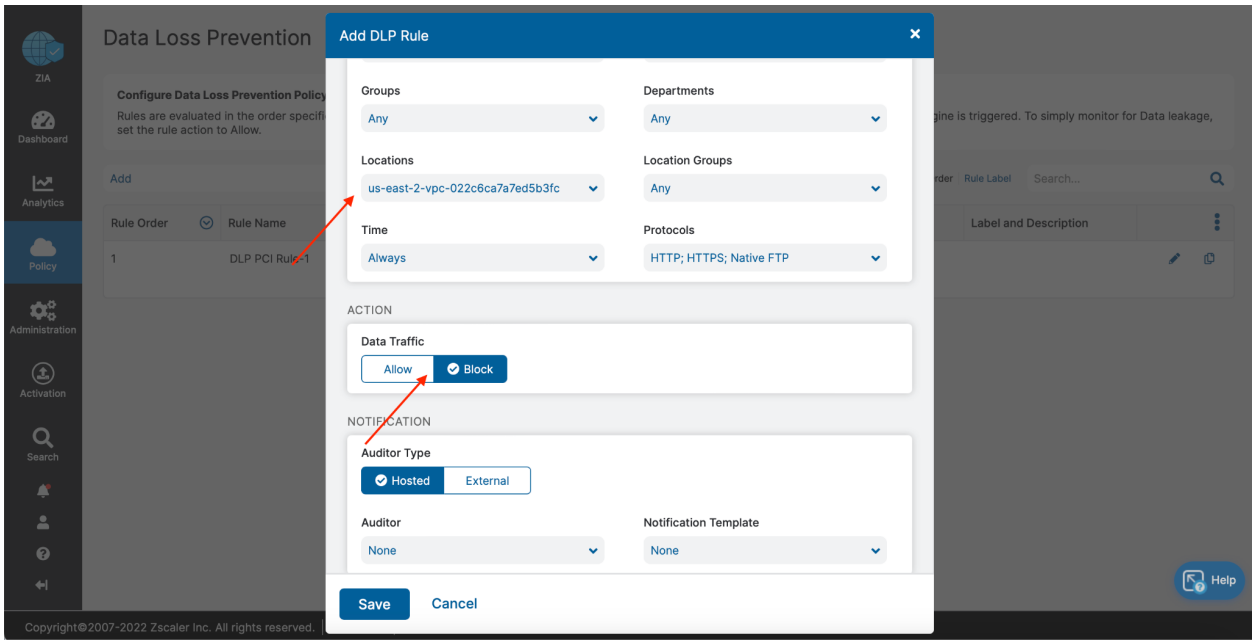
Log in to the ZIA portal and go to the Policy menu, followed by Data Loss Prevention. Click the Add button, followed by Rule with Content Inspection:



Provide a name and, for DLP Engines, select GLBA, HIPAA, and PCI, then click Done:

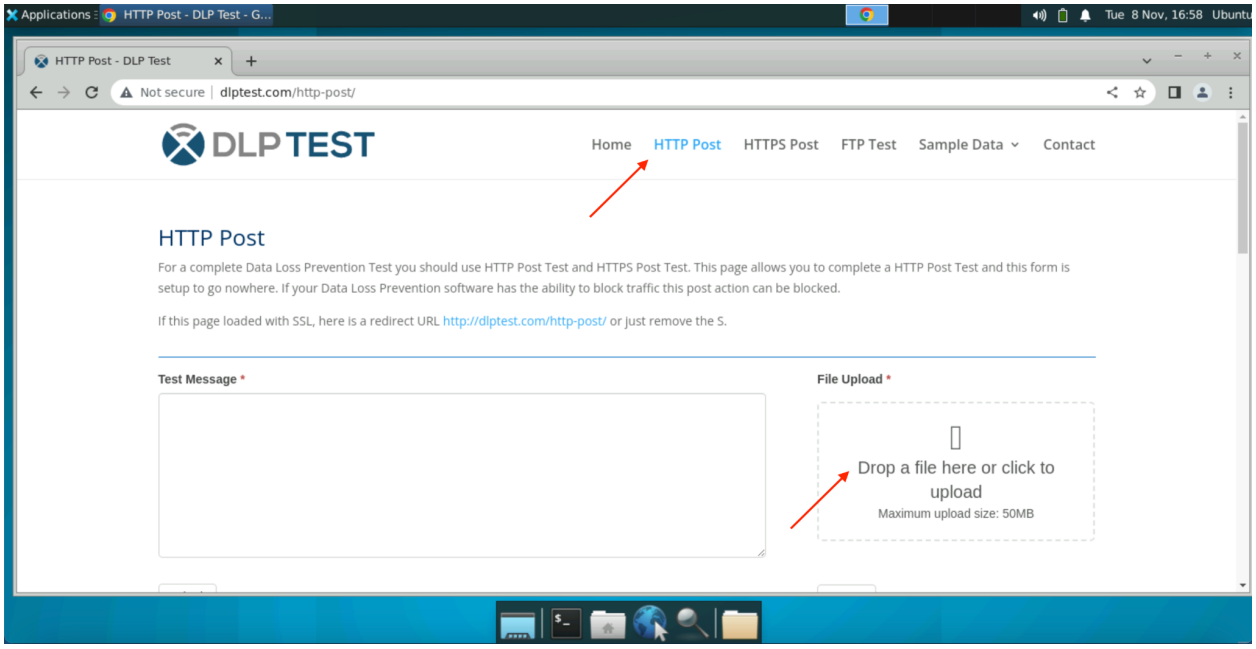


For Locations, select your USEast Location from the dropdown menu and click Done. Under the Action section, select Block. Your screen should look similar to the following:



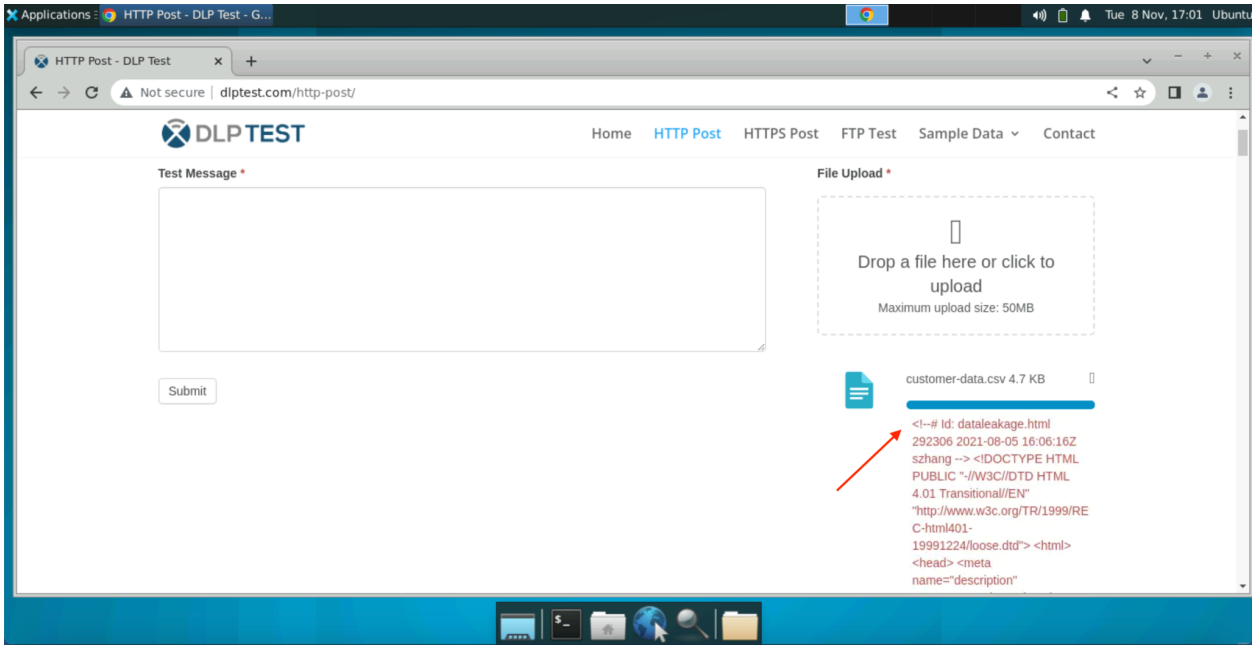
Click the Save button and activate your changes via the Activation menu.

Return to the USEast (Region2) Workload. Open an internet browser and navigate to the website https://www.dlptest.com. Click on the HTTP or HTTPS Post option at the top of the website:



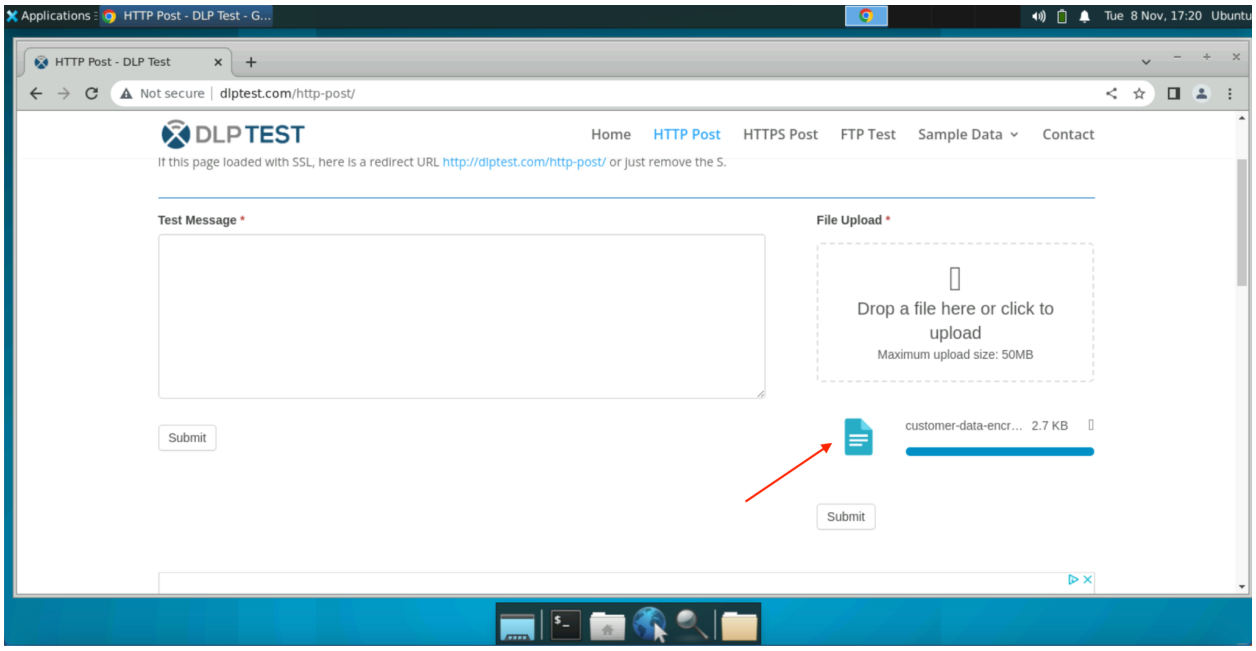
NOTE: If you wish to test an HTTPS upload, ensure your SSL Inspection policy is activated from the previous task.

Click the button to upload a file on the right and select the customer-data.csv file you reviewed earlier. Click the Upload button and notice that the file is immediately blocked from being uploaded:

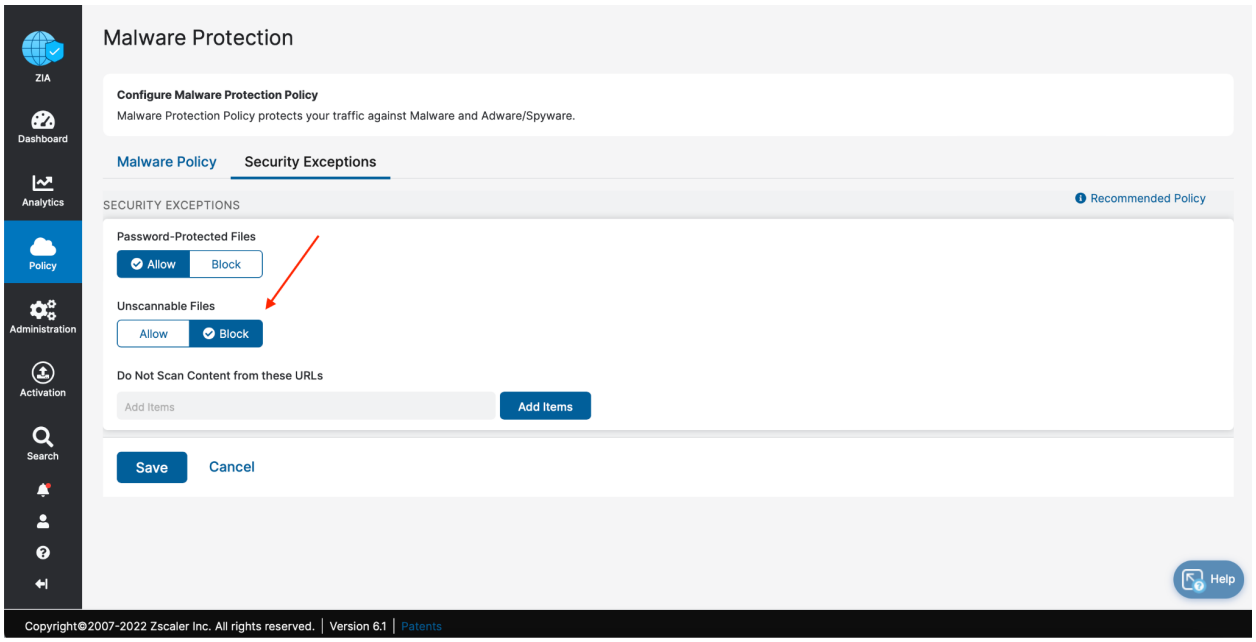


What if, however, the attacker encrypts the data prior to uploading? To simulate this, we've already encrypted the customer-data.csv file using GNU Privacy Guard (GPG). Notice the file in

the Home directory labeled customer-data-encrypted.txt. This file is the encrypted version of the file you just attempted to upload. Try uploading this file to the https://www.dlptest.com website:



It worked! Now what? How can we block this traffic? To block this traffic, let's configure ZIA to deny file uploads for unscannable files. From the ZIA portal, navigate to the Policy menu, followed by Malware Protection, then to Security Exceptions (tab). For Password-Protected Files and Unscannable Files, choose the Block option:

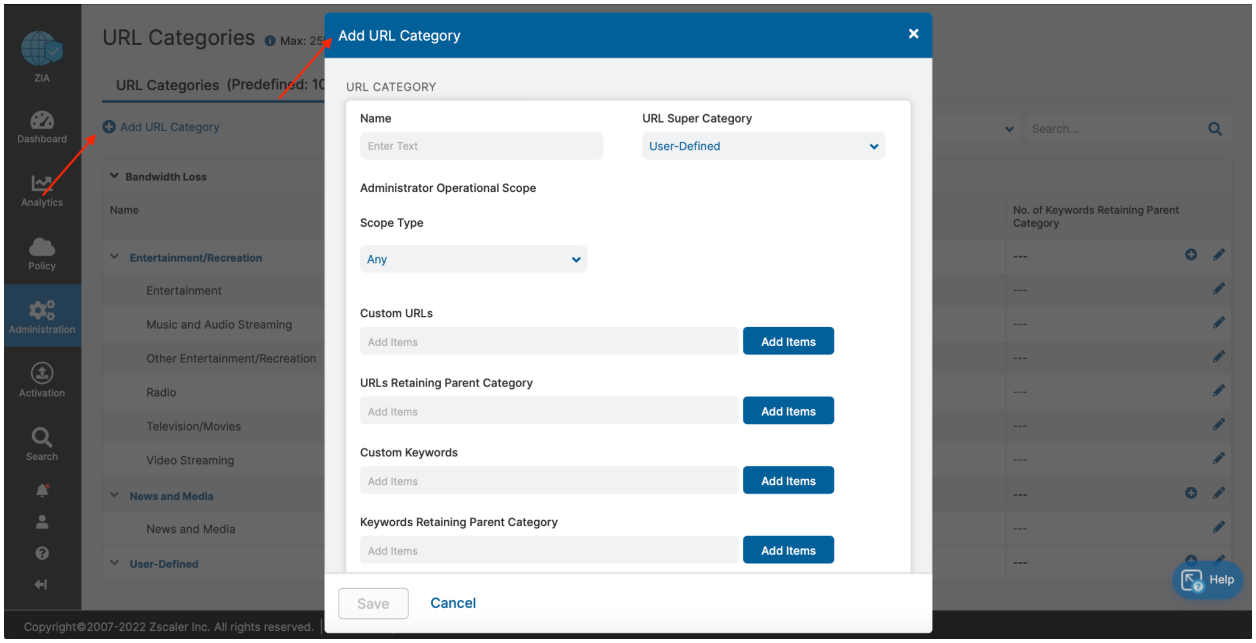


Click the Save button and Activate your change. This time, the file should fail to upload.

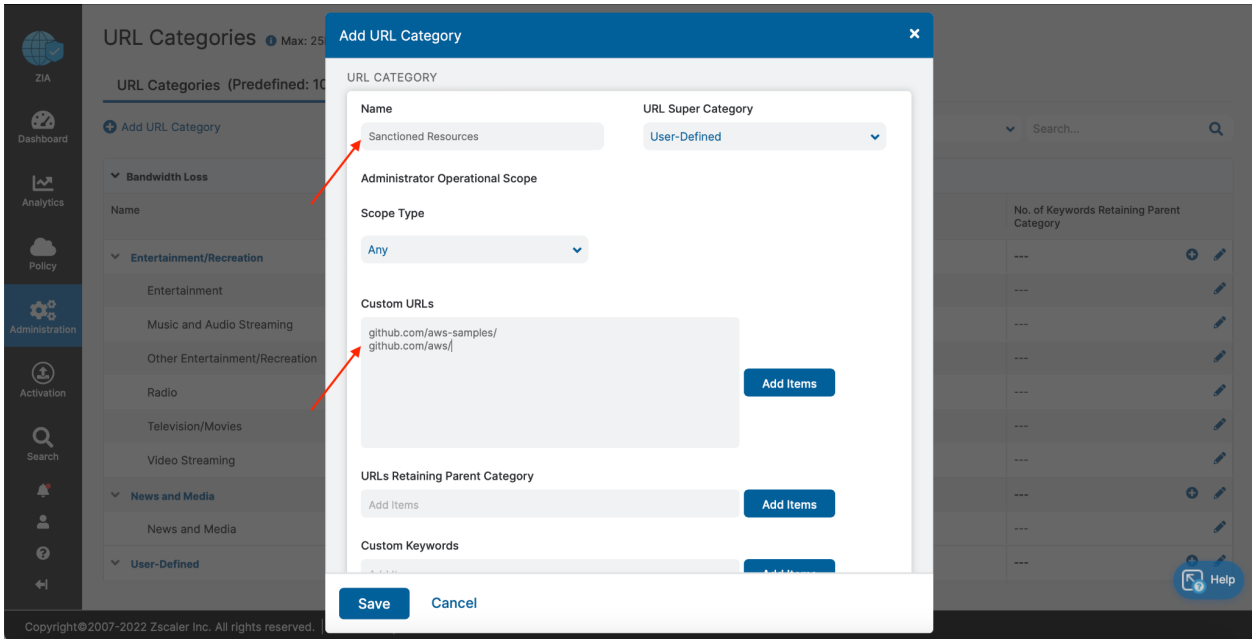
Controlling Access to Specific Resources on Websites

A company's security policy may dictate access is only granted for approved portions of a website. For example, Github contains numerous code repositories (repos), but it may not be ideal to allow unfettered access to all repos. In this lab, you'll configure the system to allow access to two official Github repos – namely github.com/aws and github.com/aws-samples.

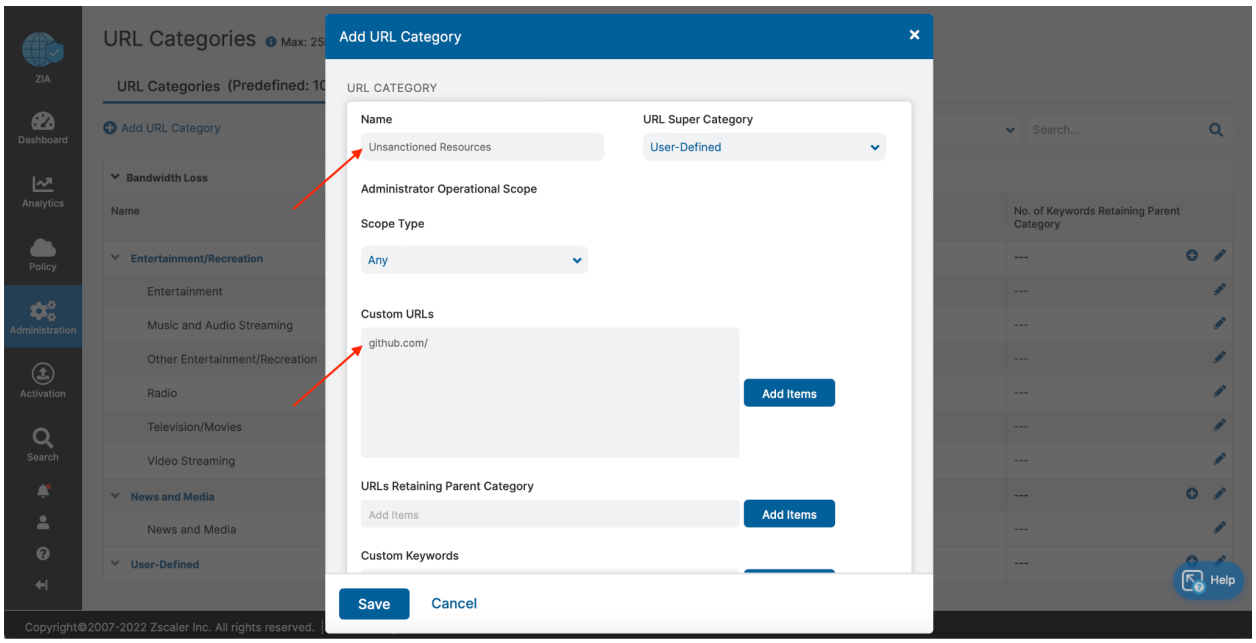
From the ZIA portal, navigate to the Administration menu, then URL Categories. Click on Add URL Category:



For Name, enter *Sanctioned Resources*. Under Custom URLs, enter the following (make sure to include the trailing "/"): *github.com/aws-samples/* and *github.com/aws/*:

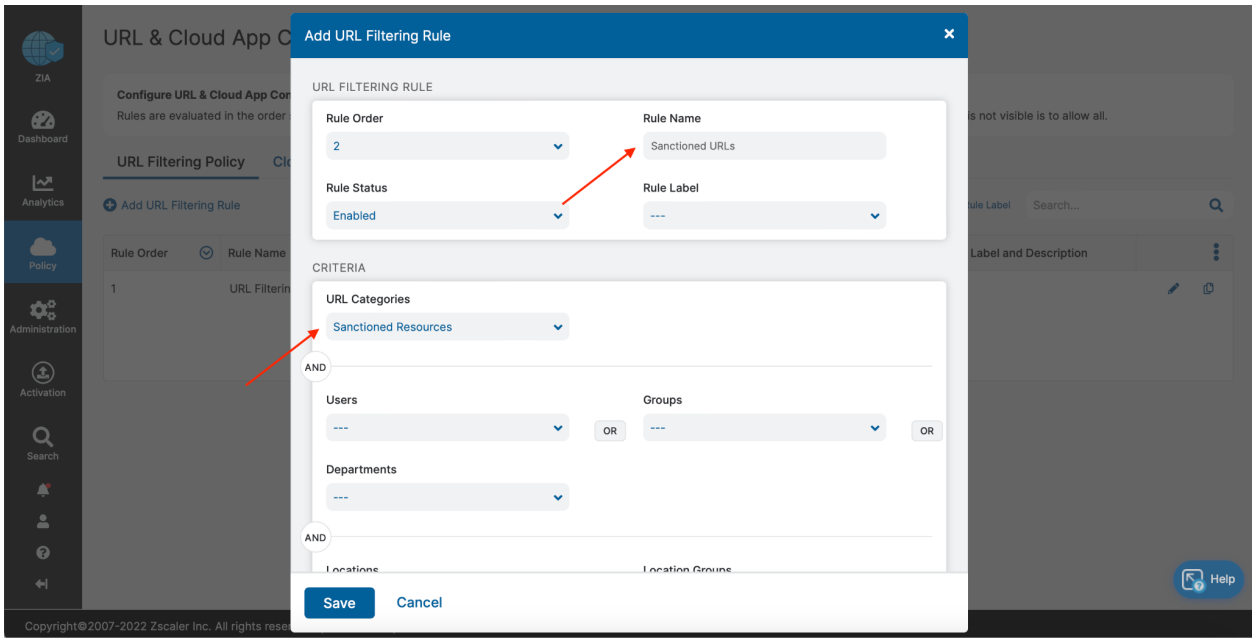


Click the Save button. Then, click the Add URL Category button again. For Name, enter *Unsanctioned Resources*. Under Custom URLs, enter the following (make sure to include the trailing "/"): *github.com/*:



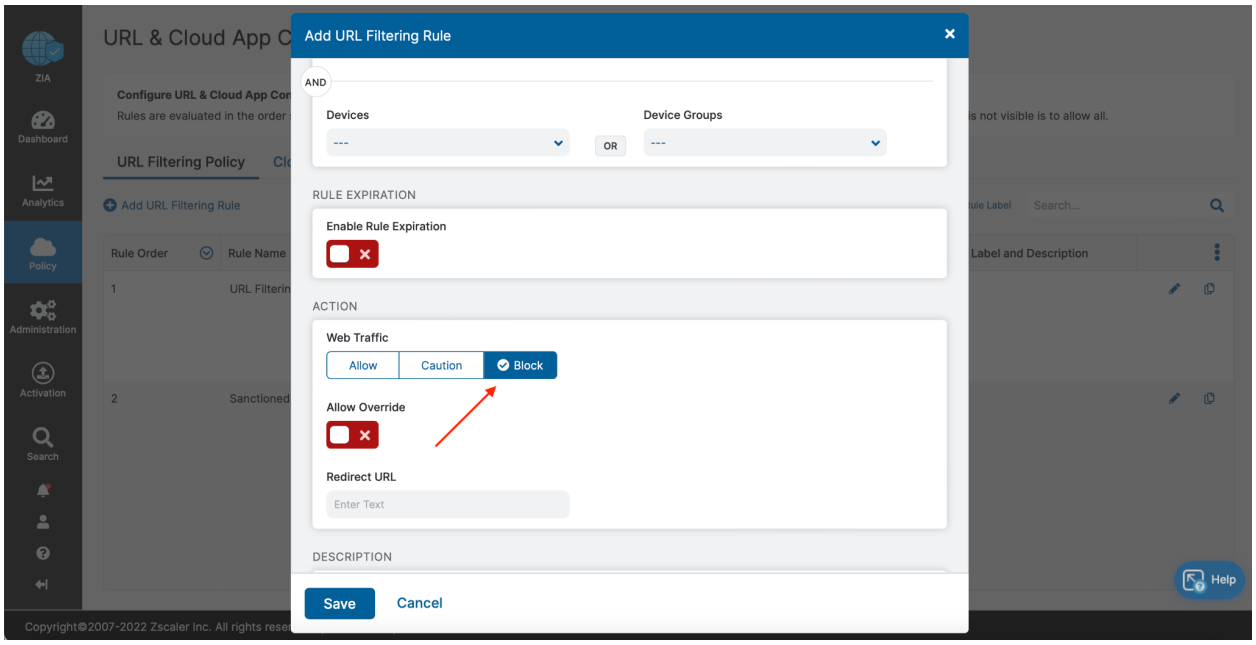
Click the Add Items button, followed by the Save button to activate your changes.

After defining the two URL categories, it's necessary to add URL Filtering rules that leverage the two categories. Navigate to Policy, then URL & Cloud App Control. First, create an allow rule for the sanctioned resources by clicking on the Add URL Filtering Rule button. Provide a name (such as Sanctioned Resources), then select the Sanctioned Resources URL Category:



In the Locations dropdown, select your USEast (Region2) Location. Then, Save and activate your change.

Similarly, add a new URL Filtering rule for unsanctioned resources that uses the Unsanctioned Resources URL Category and the same Location. In the Action section, select Block:



Save and activate your changes.

NOTE: SSL Inspection is required since GitHub will redirect HTTP requests to HTTPS. Ensure your SSL Inspection policy is enabled.

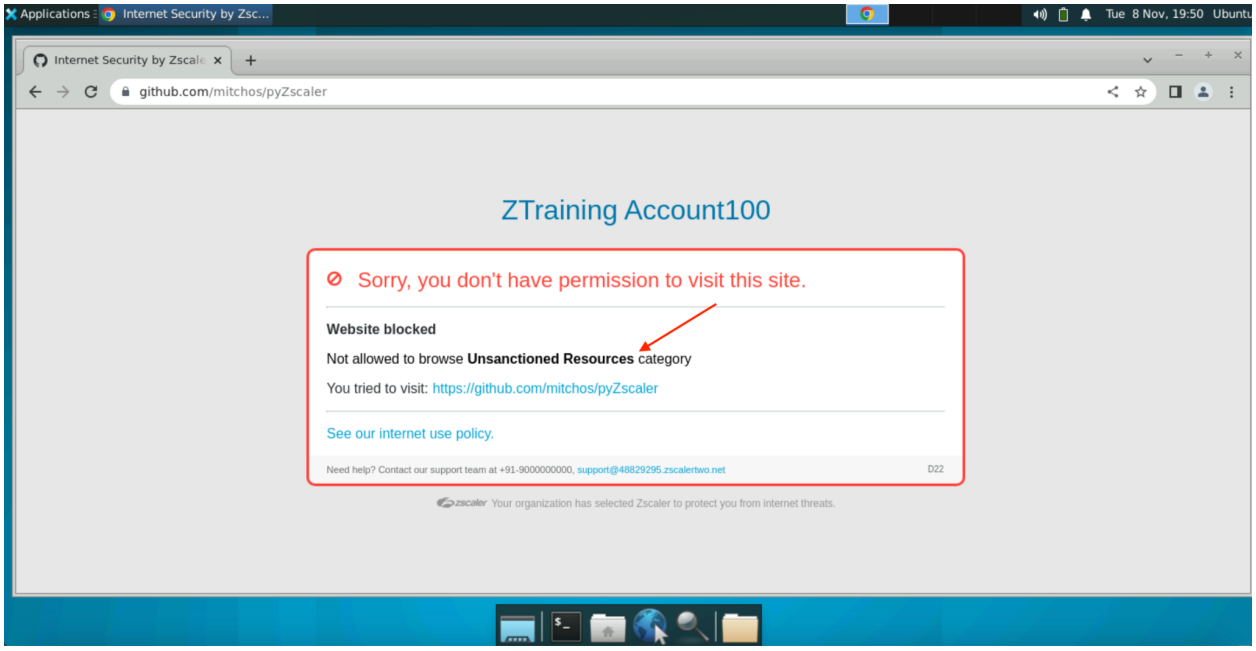
Return to the Guacamole UI and access the Region2 Workload (SSH) connection. Issue the following command: *git clone* https://github.com/mitchos/pyZscaler:

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Nov  8 19:47:55 UTC 2022

  System load:  0.080078125       Processes:             164
  Usage of /:   34.3% of 9.51GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for ens5: 10.1.1.106
  Swap usage:   0%


3 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Tue Nov  8 18:03:47 2022 from 10.1.101.145
cloudconnector@ip-10-1-1-106:~$
cloudconnector@ip-10-1-1-106:~$
cloudconnector@ip-10-1-1-106:~$ git clone https://github.com/mitchos/pyZscaler
Cloning into 'pyZscaler'...
fatal: unable to access 'https://github.com/mitchos/pyZscaler/': The requested URL returned error: 403
cloudconnector@ip-10-1-1-106:~$
```

You should notice that Git returns an error (403, Unauthorized). If you wish, you can also use *curl -v github.com/mitchos/pyZscaler* to simulate a web browser when accessing this URL for additional information:

```
<table id="en_US" width="100%" border="0" cellspacing="0" cellpadding="0">
<tbody><tr><td class="eu_h">
<i class="a_i"></i>
Sorry, you don't have permission to visit this site.
</td></tr>
<tr><td class="hr"><hr></td></tr>
<tr><td class="eu_co">
<b>Website blocked</b>
</td></tr>
<tr><td class="eu_co rsn">
Not allowed to browse <B>Unsanctioned Resources</B> category
</td></tr>
<tr><td class="eu_co">
You tried to visit:<div class="eu_l"><a href="http://github.com/mitchos/pyZscaler">http://github.com/mitchos/pyZscaler</a></div>
</td></tr><tr>
<td class="hr"><hr></td>
</tr>
<tr><td class="eu_co ln">
<a href="http://48829295.zscalertwo.net/policy.html">
See our internet use policy.
</a>
</td></tr>
<tr><td class="eu_co fo">
Need help? Contact our support team at +91-9000000000, <a href="mailto:support@48829295.zscalertwo.net">support@48829295.zscalertwo.net</a>
</td></tr>
<tr><td class="eu_co st">
<span class="s_img"></span>
Your organization has selected Zscaler to protect you from internet threats.
</td></tr>
</tbody></table>
<!--/locale en_US-->
</td></tr>
</tbody></table>
</div>
</div>
</body></html>
* Connection #0 to host github.com left intact
<!-- 647778 1 2 0 1667929800 192 http://github.com/mitchos/pyZscaler -->cloudconnector@ip-10-1-1-106:~$
```

Or, alternatively, if you wish you can also go to this website via Chrome to verify the behavior as well:



Next, let's clone the aws-cdk-examples and aws-cli repos: *git clone* *https://github.com/aws-samples/aws-cdk-examples* and *git clone https://github.com/aws/aws-cli*:

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Nov  8 19:51:45 UTC 2022

  System load:  0.0               Processes:             167
  Usage of /:   34.3% of 9.51GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for ens5: 10.1.1.106
  Swap usage:   0%


3 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Tue Nov  8 19:47:56 2022 from 10.1.101.145
cloudconnector@ip-10-1-1-106:~$
cloudconnector@ip-10-1-1-106:~$
cloudconnector@ip-10-1-1-106:~$ git clone https://github.com/aws-samples/aws-cdk-examples
Cloning into 'aws-cdk-examples'...
remote: Enumerating objects: 5778, done.
remote: Counting objects: 100% (1004/1004), done.
remote: Compressing objects: 100% (223/223), done.
remote: Total 5778 (delta 821), reused 819 (delta 773), pack-reused 4774
Receiving objects: 100% (5778/5778), 55.77 MiB | 27.91 MiB/s, done.
error: RPC failed; curl 56 GnuTLS recv error (-110): The TLS connection was non-properly terminated.
Resolving deltas: 100% (3039/3039), done.
cloudconnector@ip-10-1-1-106:~$ git clone https://github.com/aws/aws-cli
Cloning into 'aws-cli'...
remote: Enumerating objects: 106143, done.
remote: Counting objects: 100% (656/656), done.
remote: Compressing objects: 100% (331/331), done.
remote: Total 106143 (delta 359), reused 567 (delta 286), pack-reused 105487
Receiving objects: 100% (106143/106143), 109.72 MiB | 19.94 MiB/s, done.
error: RPC failed; curl 56 GnuTLS recv error (-110): The TLS connection was non-properly terminated.
Resolving deltas: 100% (70442/70442), done.
cloudconnector@ip-10-1-1-106:~$
```

The policy has been configured to allow the cloning of any repos under **aws-samples** and **aws**. So, this clone was successful!

NOTE: It's also possible to restrict the cloning of a specific repo by specifying the name of the repo in the policy.

This concludes the first lab – Cloud Connector Overview and Protecting Cloud Workloads with ZIA.

Final Thoughts

Connecting workloads to the internet across different networks is difficult. What makes this even more difficult is the traditional approach used by organizations to solve this challenge - such as technologies like VPNs and firewalls. While the outcome of connecting these workloads is achieved, the cost to achieve these goals is significant:

- Risk of lateral threats and internet-based attacks by over-extending the trusted network across the internet using VPN and WAN technologies
- Complexity increases because of complicated route filtering, multiple network hops, and fragmented policy management.
- Poor visibility across application connectivity paths and increased network blind spots.
- Increased costs due to overprovisioning network services and the use of virtual appliances such as firewalls, IPS, routers, and other point products in cloud environments.
- Limited scale and performance from the increase in network and security services used in cloud environments.

As a result, there is a need for a better approach to secure app-to-app and app-to-internet communications within multi-cloud environments. Zscaler Cloud Connector is a cloud-native zero trust access service that provides fast and secure app-to-app, app-to-internet connectivity across multi-cloud environments. With integrated, automated connectivity and security, it reduces complexity and cost and provides a faster, smarter, and more secure alternative to legacy network solutions.