Zscaler for Workloads (EDU-240)

Hands-on Lab Guide

Zscaler for Workloads (EDU-240) Lab Guide
Feb. 2024, Rev. 1.0a

Contents

About the Zscaler for Workloads Hands-on Labs

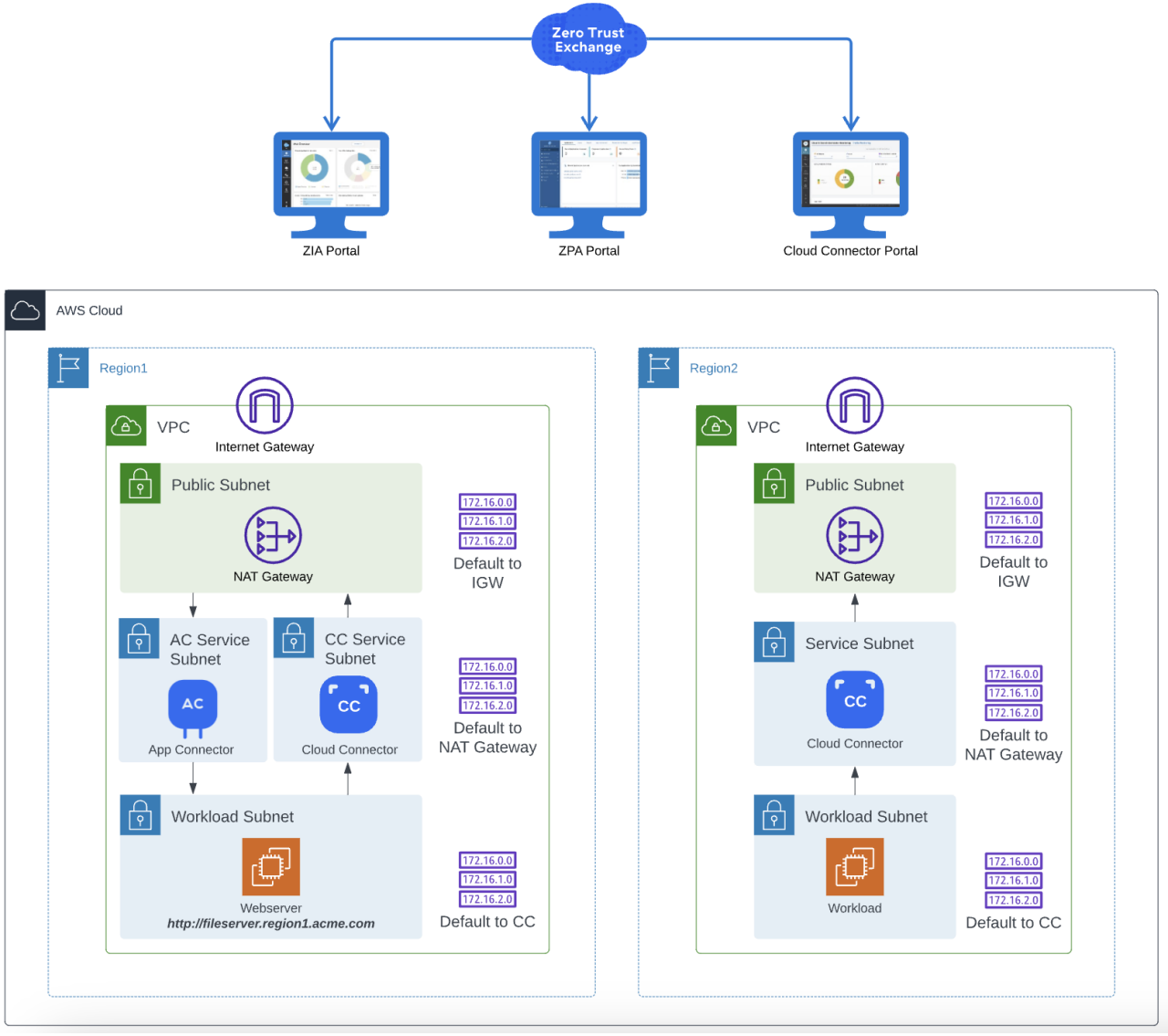
Keeping applications and cloud workloads secure is a growing concern across all industries. Zero trust architecture is an effective approach to workload security. Zscaler Workload Communications on Amazon Web Services (AWS) allows you to securely connect your applications anywhere by minimizing the attack 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 lab covers three main areas:

- Cloud Connector Overview
- Protecting Workload Internet Access with ZIA
- Integrating with Zscaler Private Access

Lab Setup

This lab deploys resources in two AWS regions (Region1 and Region2 in the diagram).



Accessing the Environment

Your pod has been configured with a Bastion host to proxy communication between your machine and each of the lab components you wish to access without the need for a VPN or client. As part of the event, a **URL was displayed that you can use to secure a pod**. Simply browse to this URL, input your information and click the **RSVP** button.

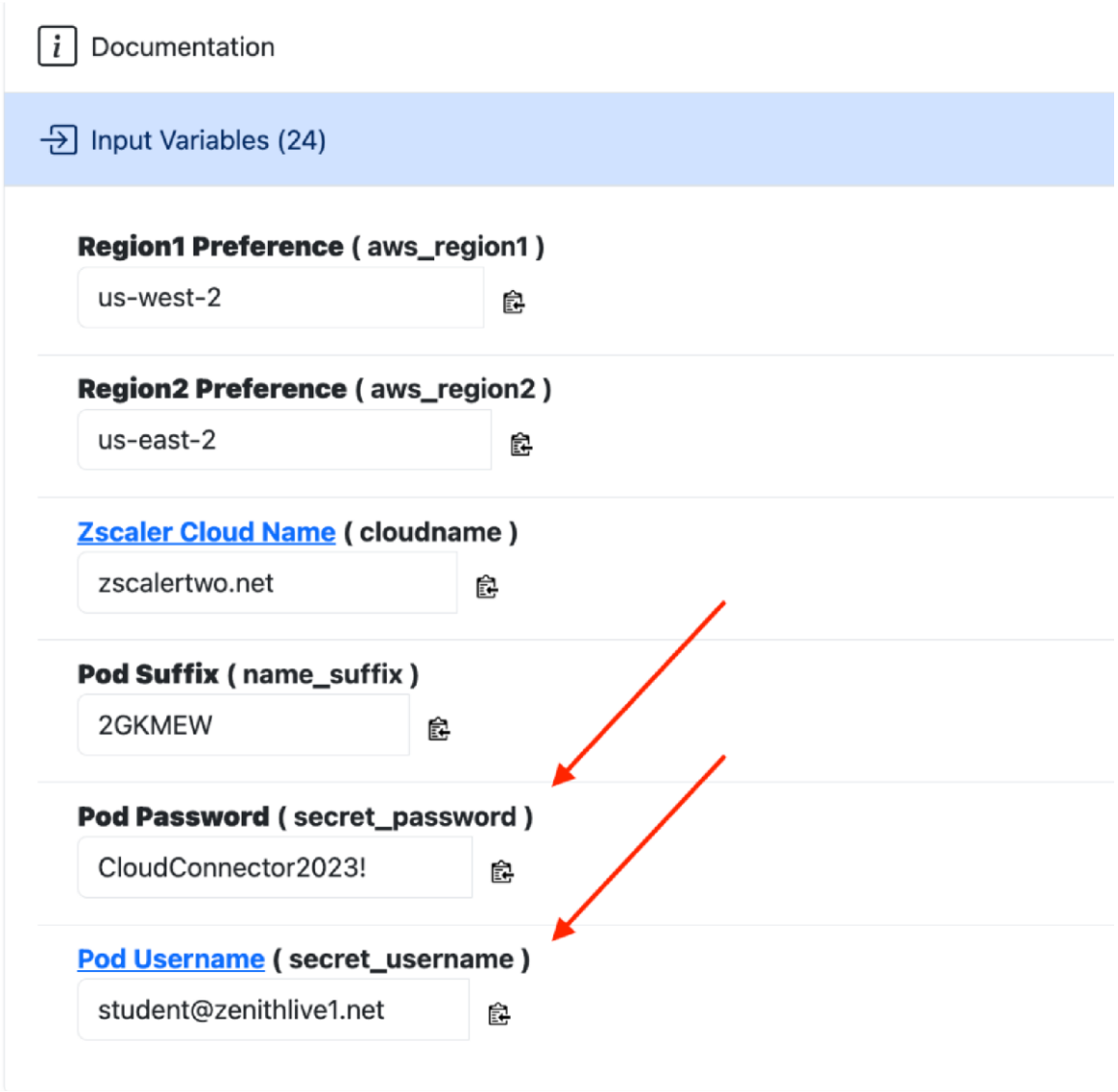
Task 1: Connect to your pod

In this task, you will connect to your lab environment. To connect to your pod and log in, follow these steps:

1. Navigate to the **RSVP URL** displayed during your session.
2. Enter the information requested.
3. Click **RSVP!**.

Note: *Once registered, the demo tool will place you directly into your pod's dashboard. You can access any of your pod's resources by simply clicking on the icons in the image map. Clicking each will open a new tab with access into that respective device, so* ***ensure pop-up blockers are disabled****.*

You can find the credentials for your pod at the bottom of your pod dashboard under the **Input Variables** dropdown. These will be used to login to the Cloud Connector and ZIA/ZPA portals, respectively:

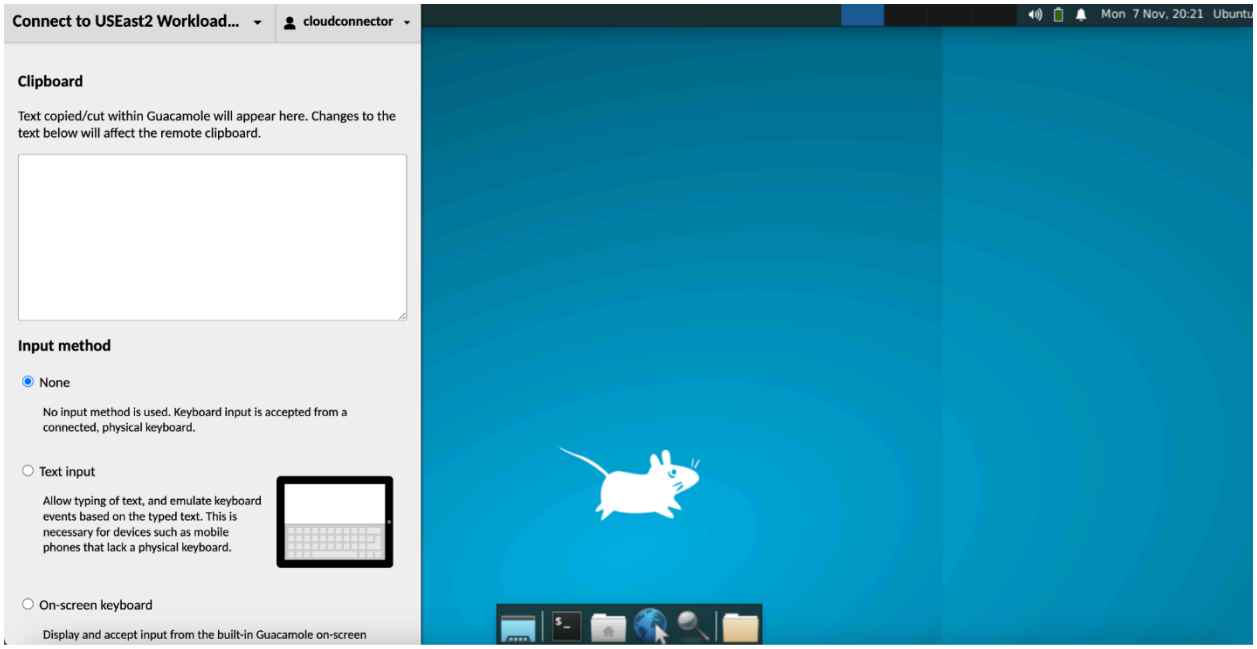


For reference, the table below lists the credentials to all pods (replace 'XX' in the username with your pod's number):

	Username	Password
ZIA Portal	student@zenithlive*XX*.net	CloudConnector2023!
ZPA Portal	student@zenithlive*XX*.net	CloudConnector2023!
Cloud Connector Portal	student@zenithlive*XX*.net	CloudConnector2023!
Test Machines	cloudconnector	CloudConnector2022!

4. To adjust settings within your current window, press **CTRL**+**ALT**+**SHIFT** (CTRL+OPTION+SHIFT on Mac).

Note: *This option is useful for split-screening consoles, copying or pasting, and quickly navigating between connections.*



Pre-Lab: Provision AWS and Azure

Your pod has been preconfigured with Cloud Connector appliances using an automation tool. Before we get started, let's review what this tool did to understand what was configured within the cloud provider. When you RSVP'd, a modal should have appeared on your screen with a click through demo of the AWS workflow:

This modal will walk you through the basic deployment of a Cloud Connector appliance inside of AWS. Take a moment to walk through this click-through demo. There are three main steps to consider when deploying the appliances within a public cloud:

1. Subscribing to the Cloud Connector AMI (or VHD, if Azure) in the Marketplace.
2. Creating a Secrets Manager object to store registration credentials (or Key Vault, if Azure) that the appliance will use to authenticate to the Zscaler service.
3. Running automation scripts to deploy the infrastructure (CloudFormation, Terraform, Marketplace ARM, etc.)

The demonstration in the modal used CloudFormation Templates to deploy the appliances. In this case, CloudFormation Templates assume a brownfield environment where basic infrastructure already exists: such as a VPC, Subnets, Route Tables, Internet Gateway, etc. Hence, the modal does not demonstrate the deployment of those resources.

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.

Lab 1: Navigating the Cloud Connector Dashboard

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.

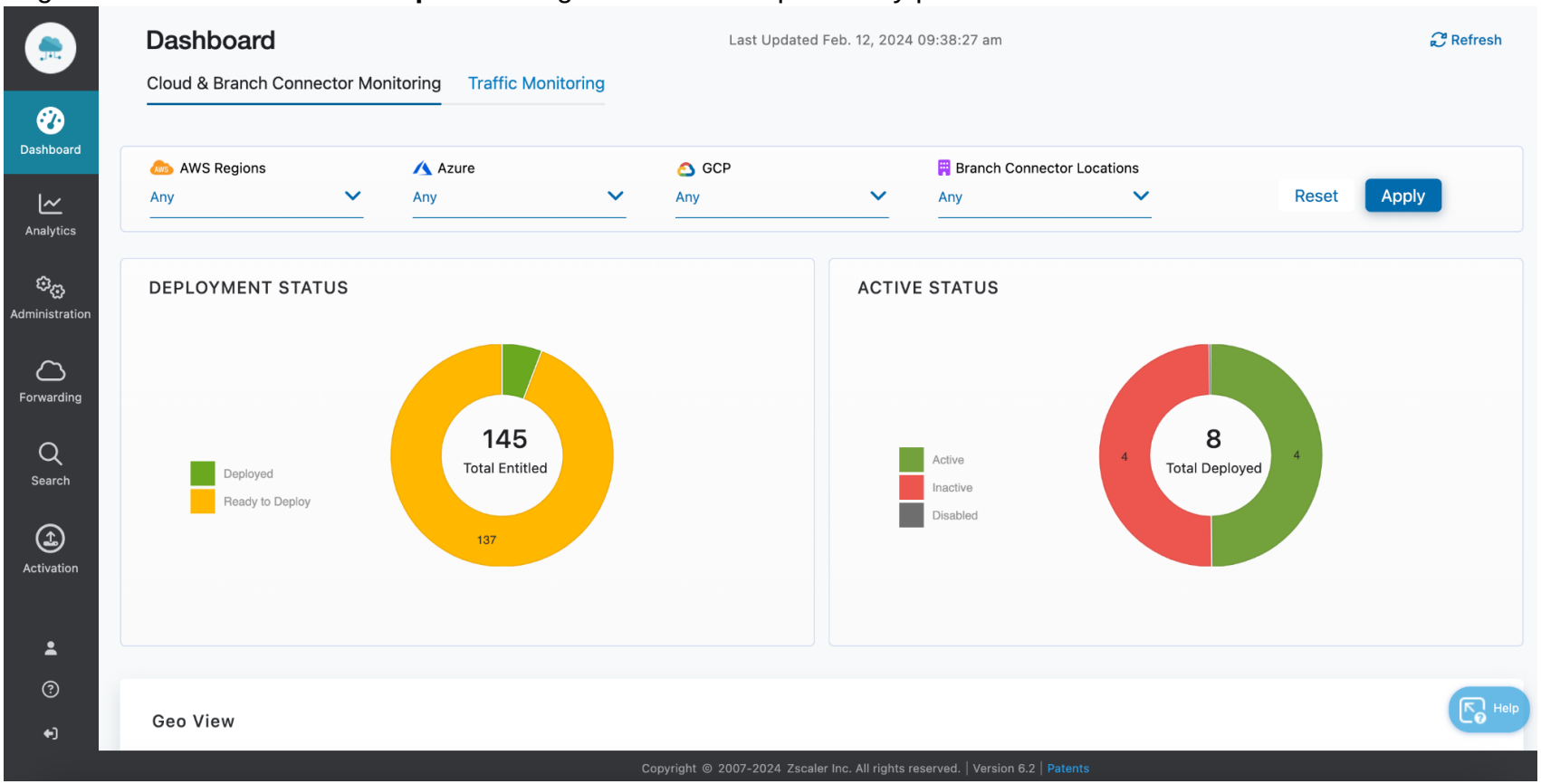
Task 1: Navigate to the Cloud Connector Dashboard

In this task, you will familiarize yourself with the Cloud Connector dashboard and its three main functions:

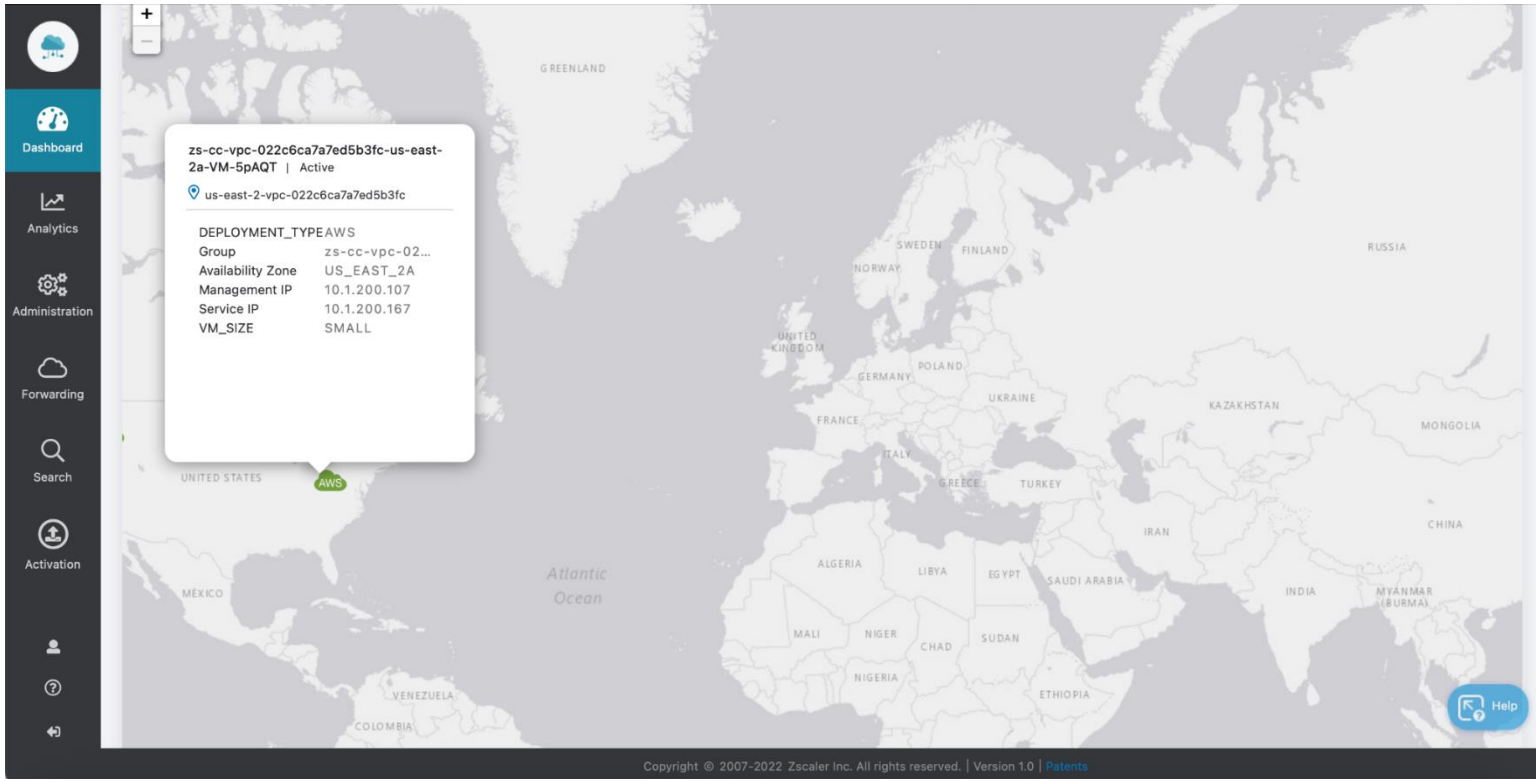
- Facilitating the onboarding, provisioning, and registration process of appliances within various Branch and Cloud environments.
- Providing a new logging perspective on traffic leaving or transiting the cloud. This can aid in troubleshooting as well as threat correlation.
- Providing a traffic steering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

Navigate to the Cloud Connector portal by clicking on the icon at the top of your pod's image map:

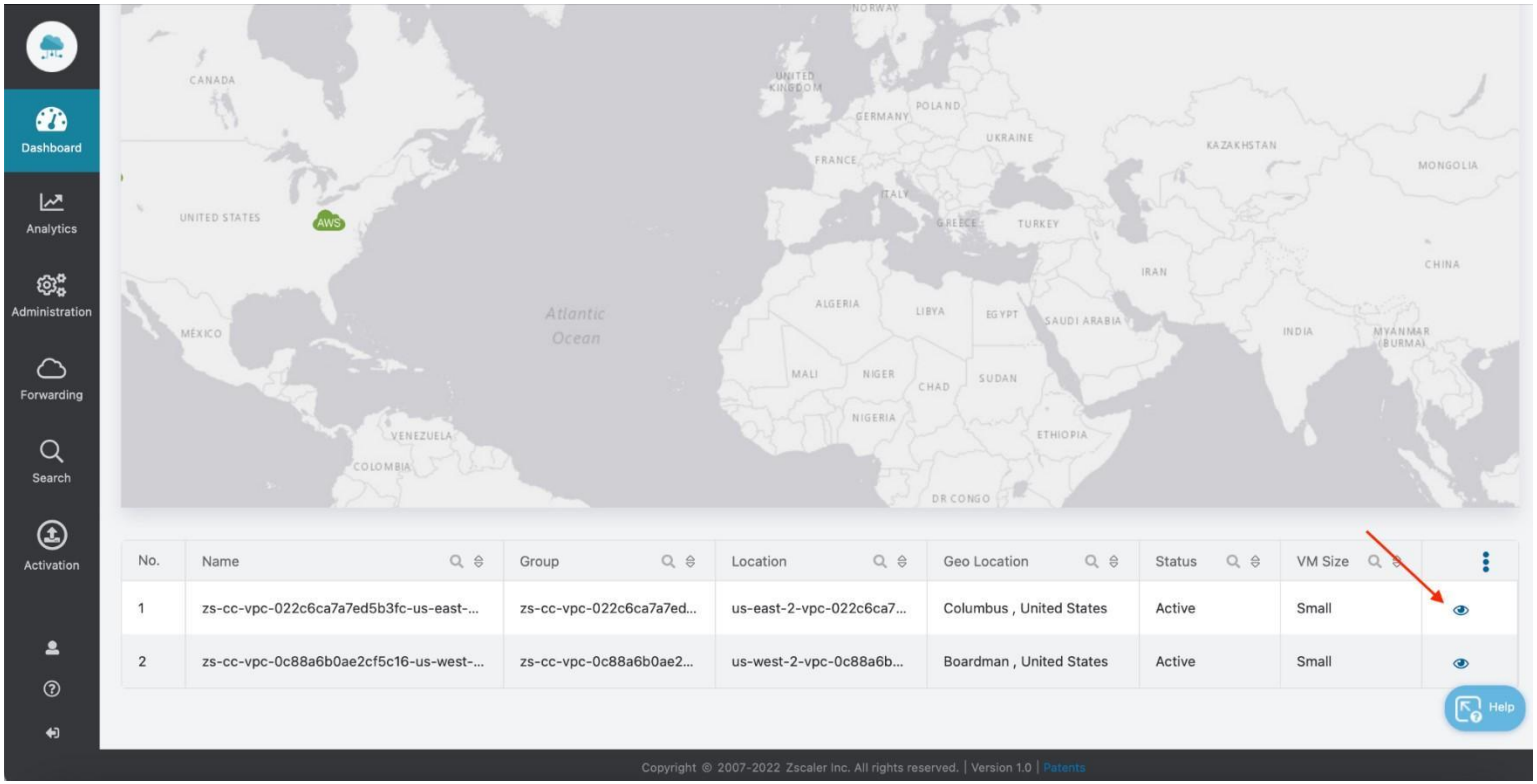
1. Log in to the **Cloud Connector portal** using the credentials previously provided.



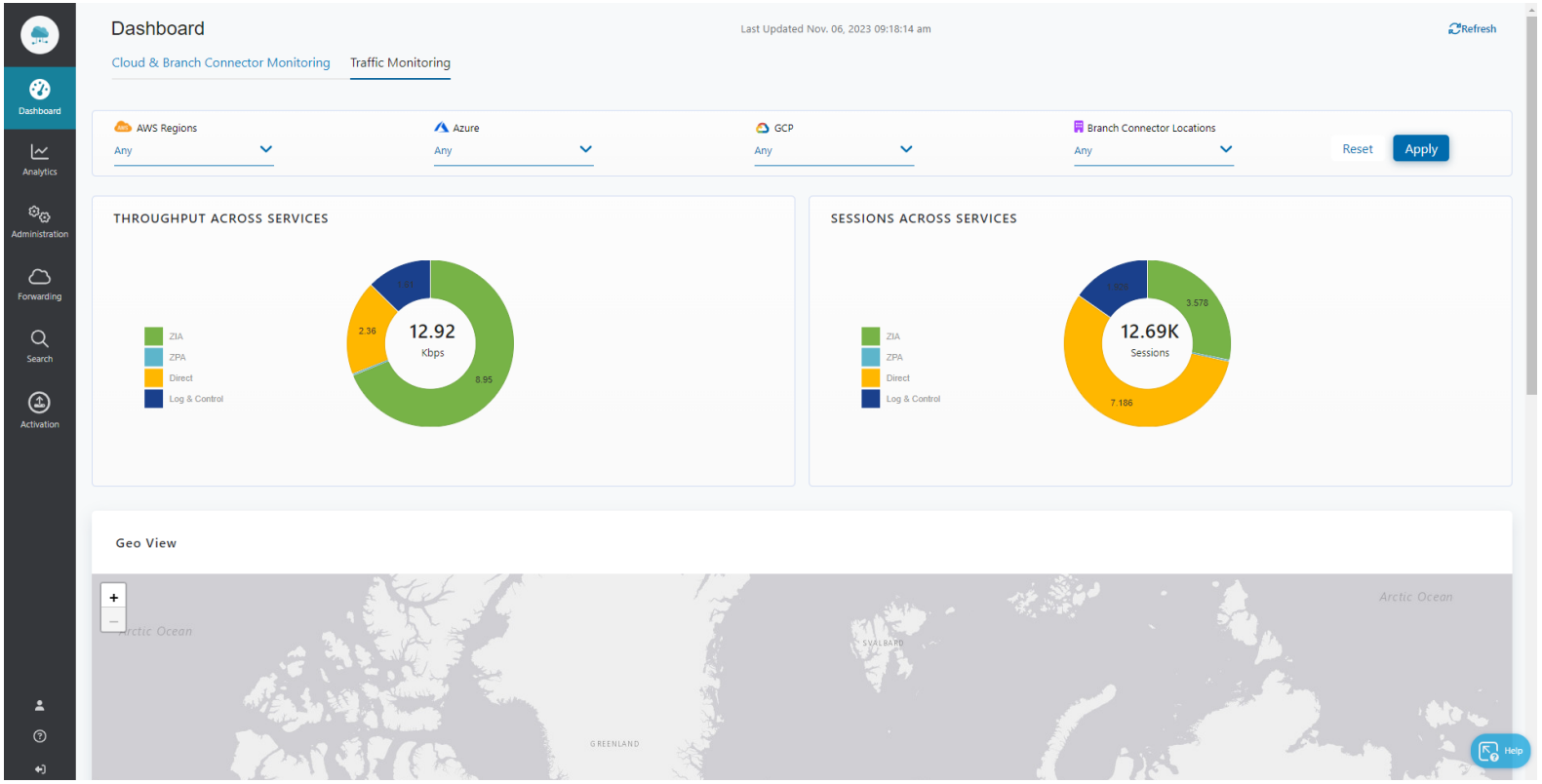
2. At the top of the dashboard, review the number of Cloud Connector appliances deployed and the number of active vs. inactive. Note that the inactive Cloud Connector appliances are normal. Your pod has been pre-configured with AWS Auto Scale functionality. The inactive appliances represent Warm Pool instances that will remain inactive until traffic demand increases. Your pod may have more inactive instances than depicted in the preceding graphic, though all pods should have at least four active instances (two in each region)
3. Scroll down to a geographical view of where each of the Cloud Connector appliances are deployed.
4. Click on the Cloud Connector icon on the map to view additional information about the appliance, including appliance size, health, status, VPC information.



5. Scroll down further to see a listing of the Cloud Connector appliances.
6. Click on the view details icon (eyeball) to the right of one of your Cloud Connectors.



7. In this screen, review additional information about the Cloud Connector appliance, such as General and Management Information and Forwarding Information.
8. In the upper left portion of the screen, click the **Traffic Monitoring** tab.



9. Review statistics about the traffic that this appliance is processing.

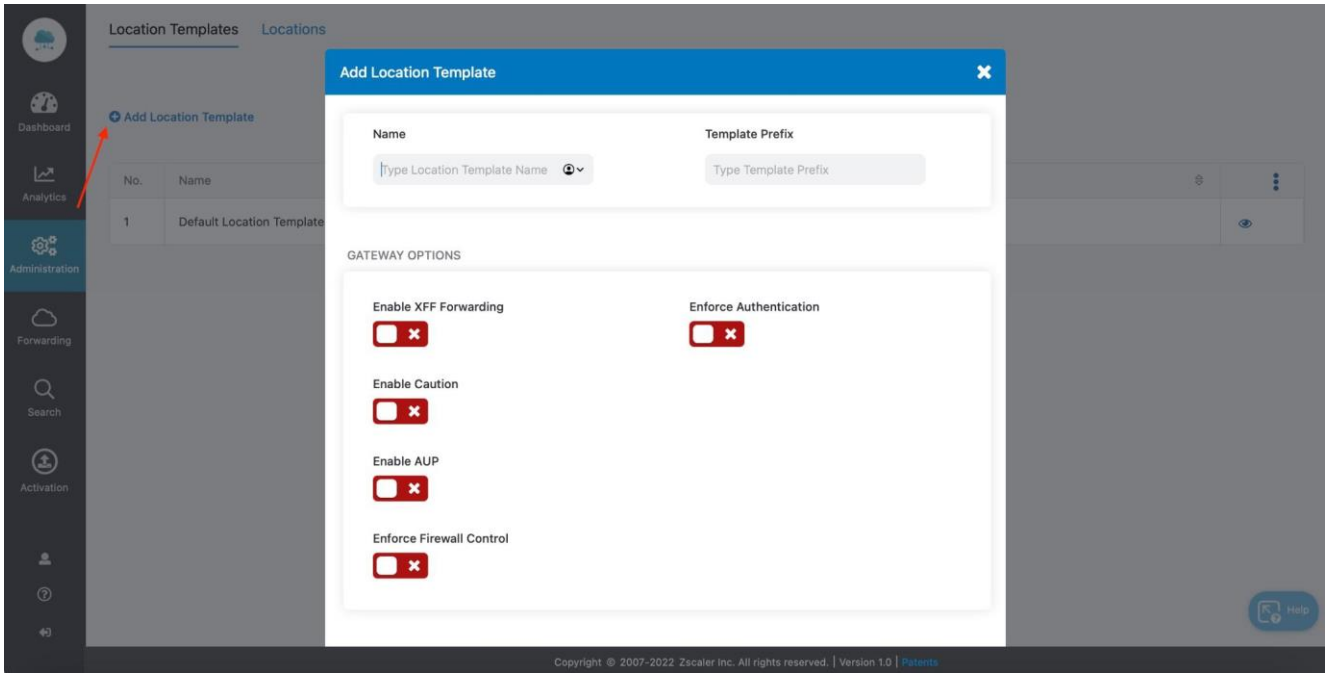
Lab 2: Managing Location and Provisioning Templates

Cloud Connector appliances use *Provisioning Templates* and *Location Templates* to bootstrap themselves when they register.

Task 1: Configure a Location Template

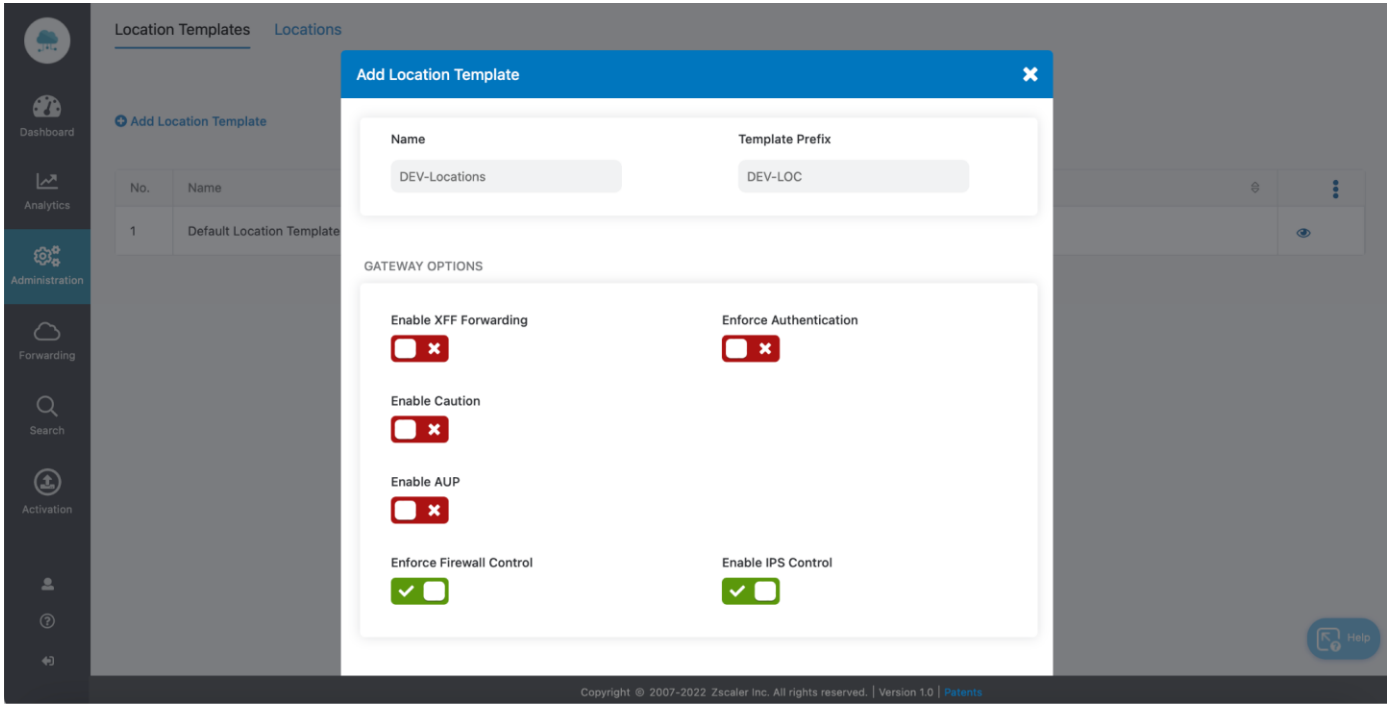
Zscaler Cloud Connector appliances automatically create Locations based on the Cloud Service Provider networks that they serve. Controlling which features are enabled or disabled for dynamically created Locations is the job of a Location Template. In this task, you will create a new Location Template.

1. In the Cloud Connector portal, select **Administration > Location Templates > Add Location Template**.



2. To configure the template:
   a. Enter a **Name**.
   b. Optionally, enter a **Template Prefix**.

Note: *The Template Prefix will be prepended to all Locations this template is attached to in order to help make a Location more easily identifiable.*



   c. Select the options you wish to enable.
   d. Click **Save**.

Note: *This template will not be used, so feel free to explore.*

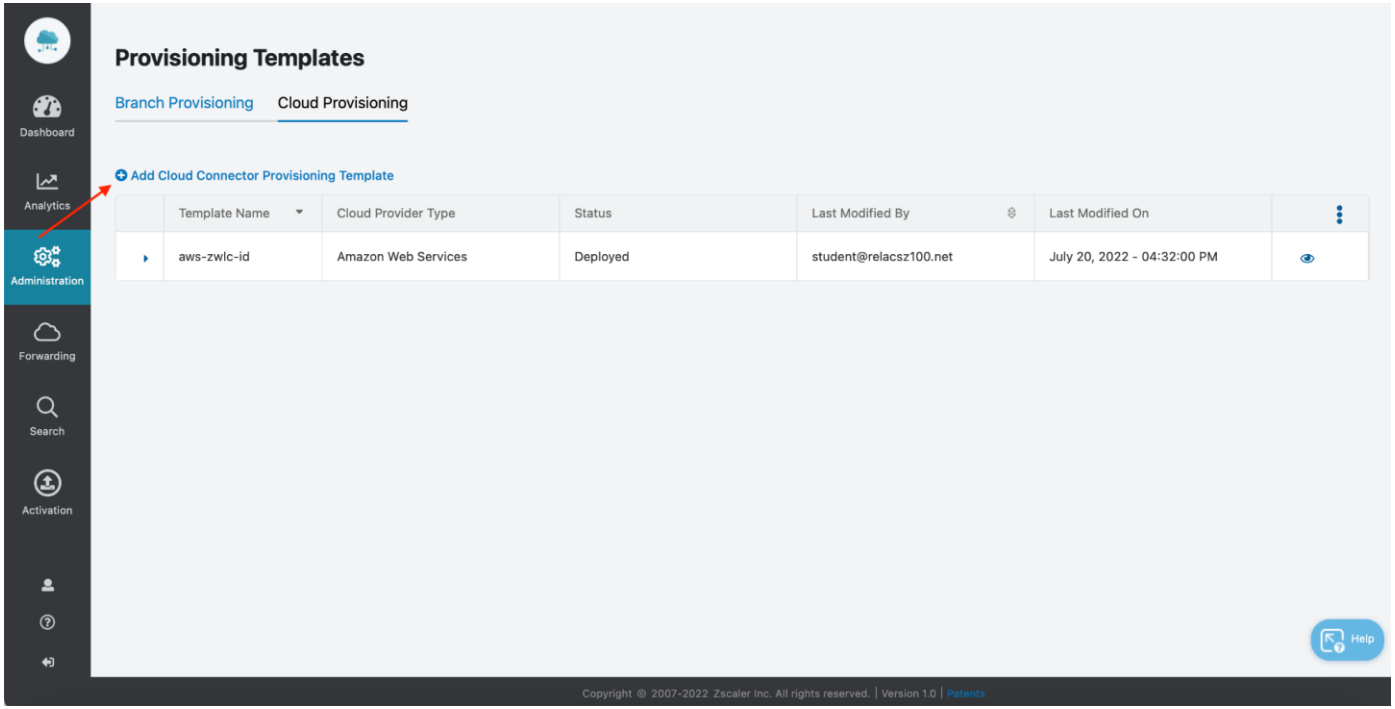
Task 2: Configure a Provisioning Template

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

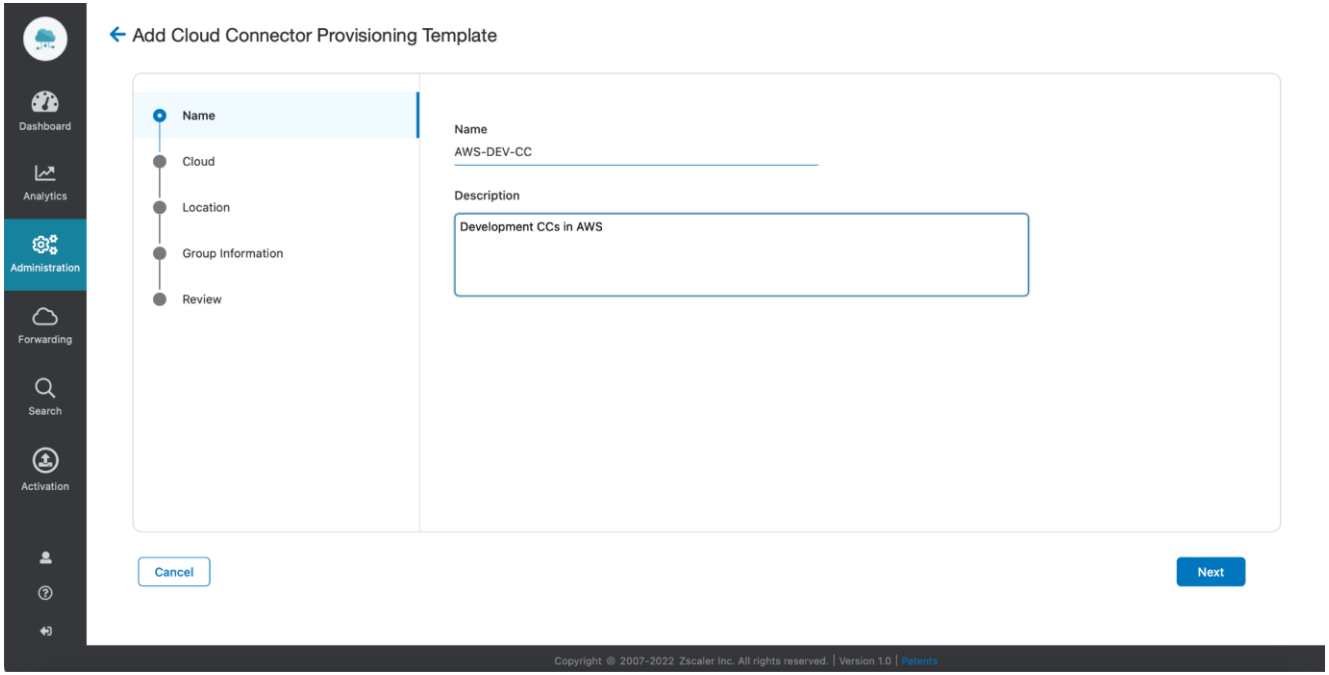
1. In the Cloud Connector portal, select **Administration > Provisioning & Configuration**.

Note: *An AWS template has already been configured for you. In fact, your currently registered Cloud Connectors used this template when registering.*

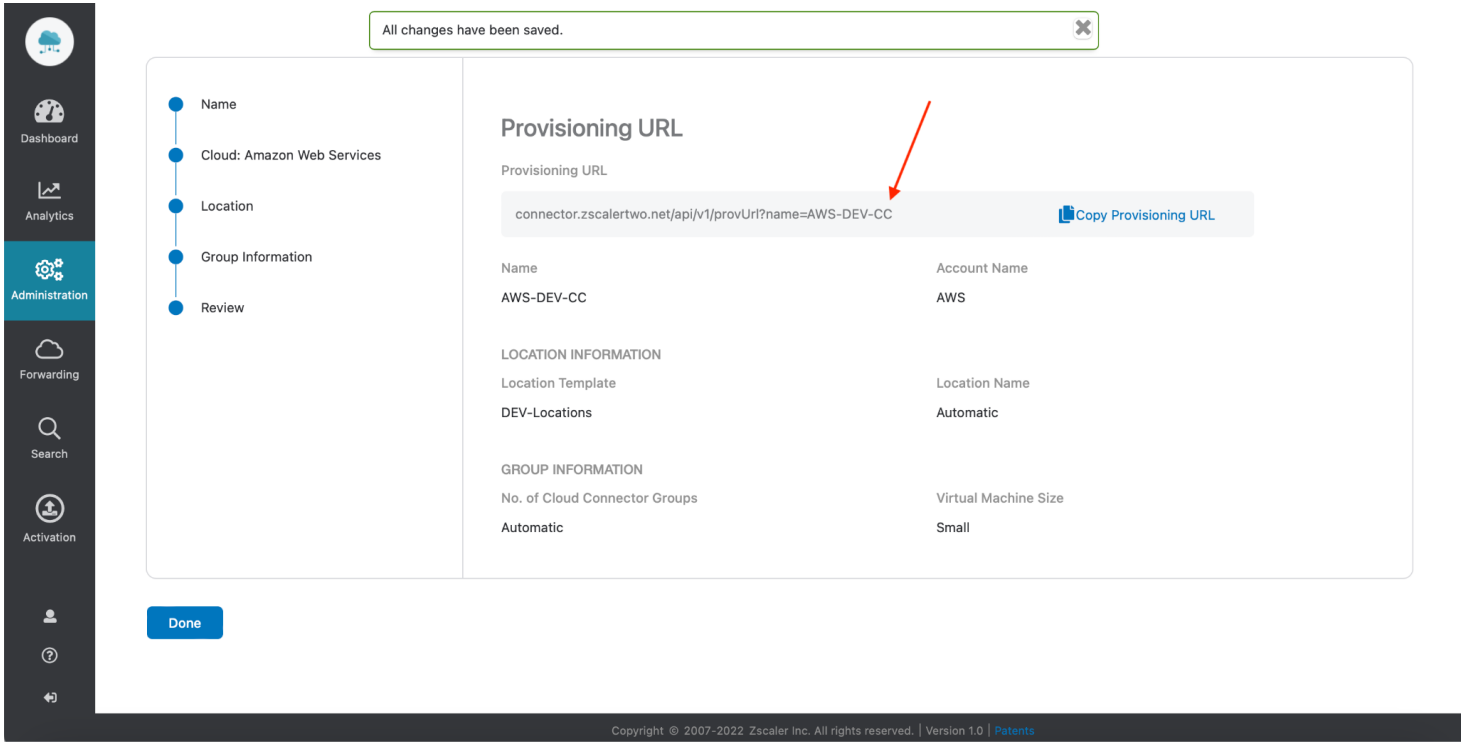
2. Click **Add Cloud Connector Provisioning Template**.



3. To configure the template:
   a. Enter a **Name**.
   b. Enter a **Description**.



c. Click **Next**.
d. In the Location Template dropdown menu, select the **Location Template** you created in the previous steps.
e. Click **Next**.
f. Leave the VM size as **Small**, and click **Next.**
g. Complete the workflow by clicking **Save**.
h. In the screen that appears, note the **Provisioning URL**.



Note: *The Provisioning 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.*

i. Click **Done**.

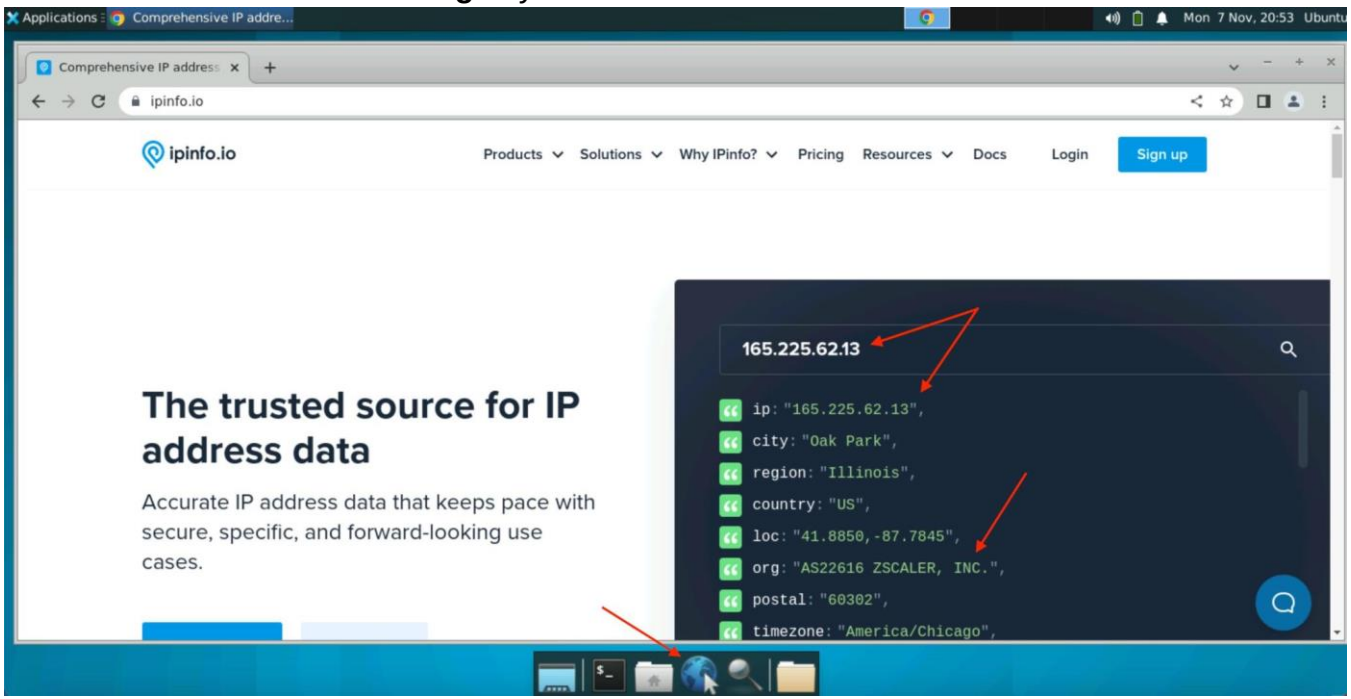
Lab 3: Managing Traffic Forwarding Policy

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

Task 1: Verify Internet Connection

In this task, you will check how one of your workloads' currently connects to the Internet.

1. From your pod dashboard, access your **Region1** Workload by clicking on it.
2. In the console of your machine, open a web browser and browse to **https://www.ipinfo.io**.
3. Note the **IP Address** and **Org** of your workload.



Note: *This traffic reached the Internet via ZIA and used one of Zscaler's public IP addresses.*

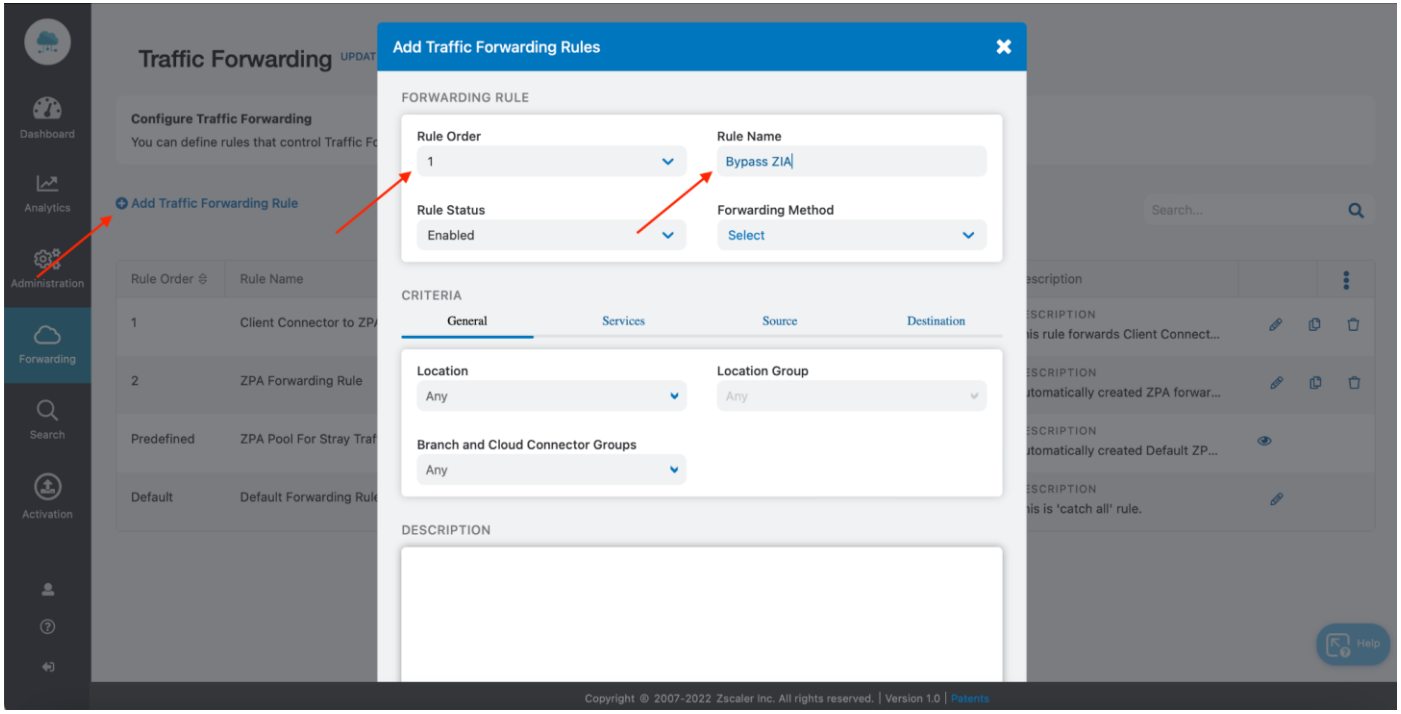
Task 2: Configure a Traffic Forwarding Rule

Let's assume that you'd like to prevent this traffic from using ZIA and, instead, go directly out to the Internet from AWS. In this task, you will configure a Traffic Forwarding Policy and then verify its impact on a workload's traffic.

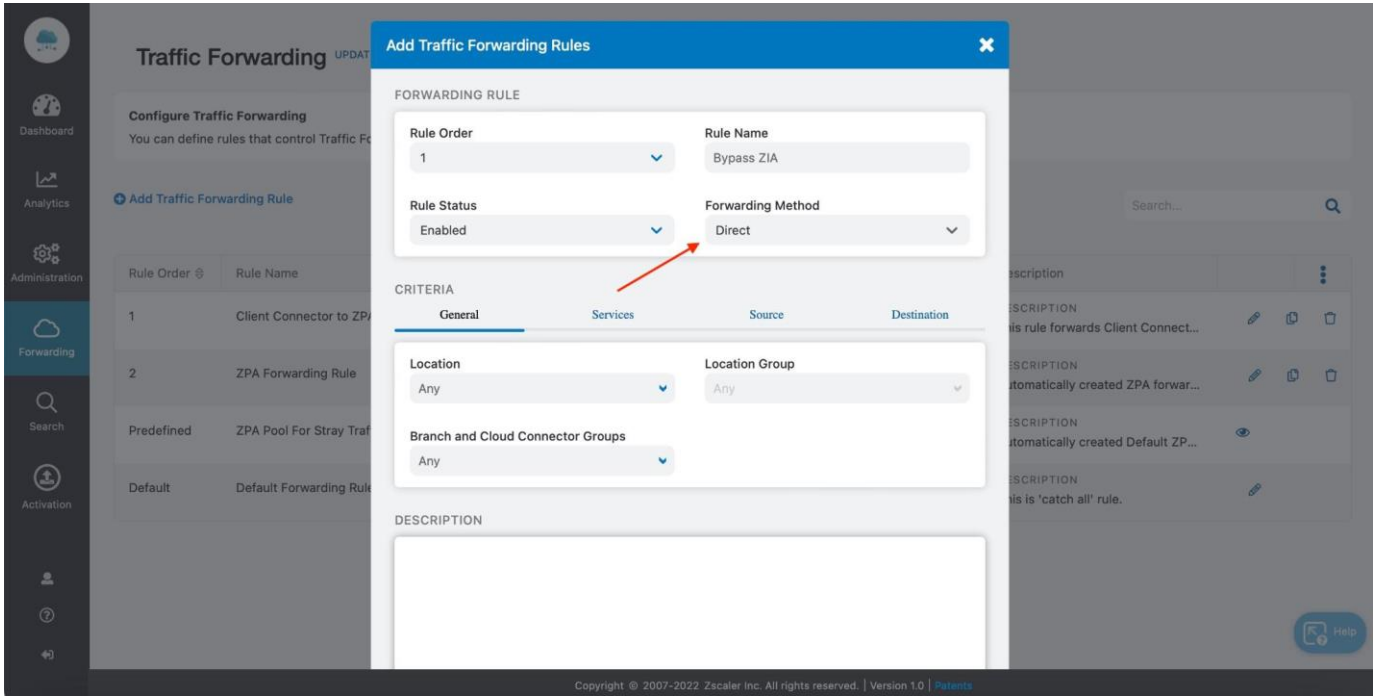
1. In the Cloud Connector portal, select **Forwarding > Traffic Forwarding**.

Note: *You will see a few default rules installed automatically when the Cloud Connector registered. These rules triggered when you sent traffic to https://www.ipinfo.io (specifically, the Default ZIA rule).*

2. Click **Add Traffic Forwarding Rule**.
3. Configure the rule as follows:
   a. Change the Rule Order to **1** to ensure your new rule is placed at the top.
   b. Enter a **Name**.



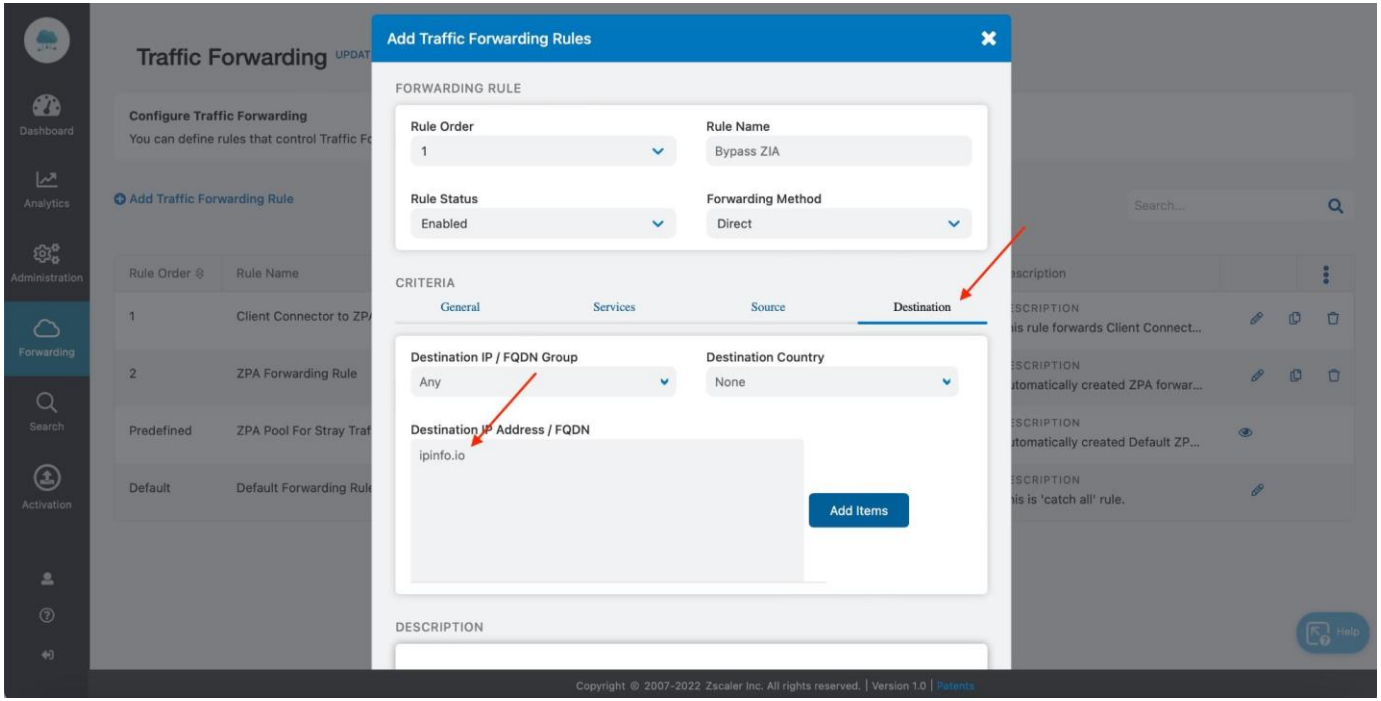
c. In the Forwarding Method dropdown, choose the **Direct** option.



Note: *In the Criteria section, you can select all of the options available for identifying which traffic will adhere to this rule.*

---

d. Click the **Destination** tab.
e. In the Destination IP Address / FQDN field, enter **ipinfo.io**.



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.

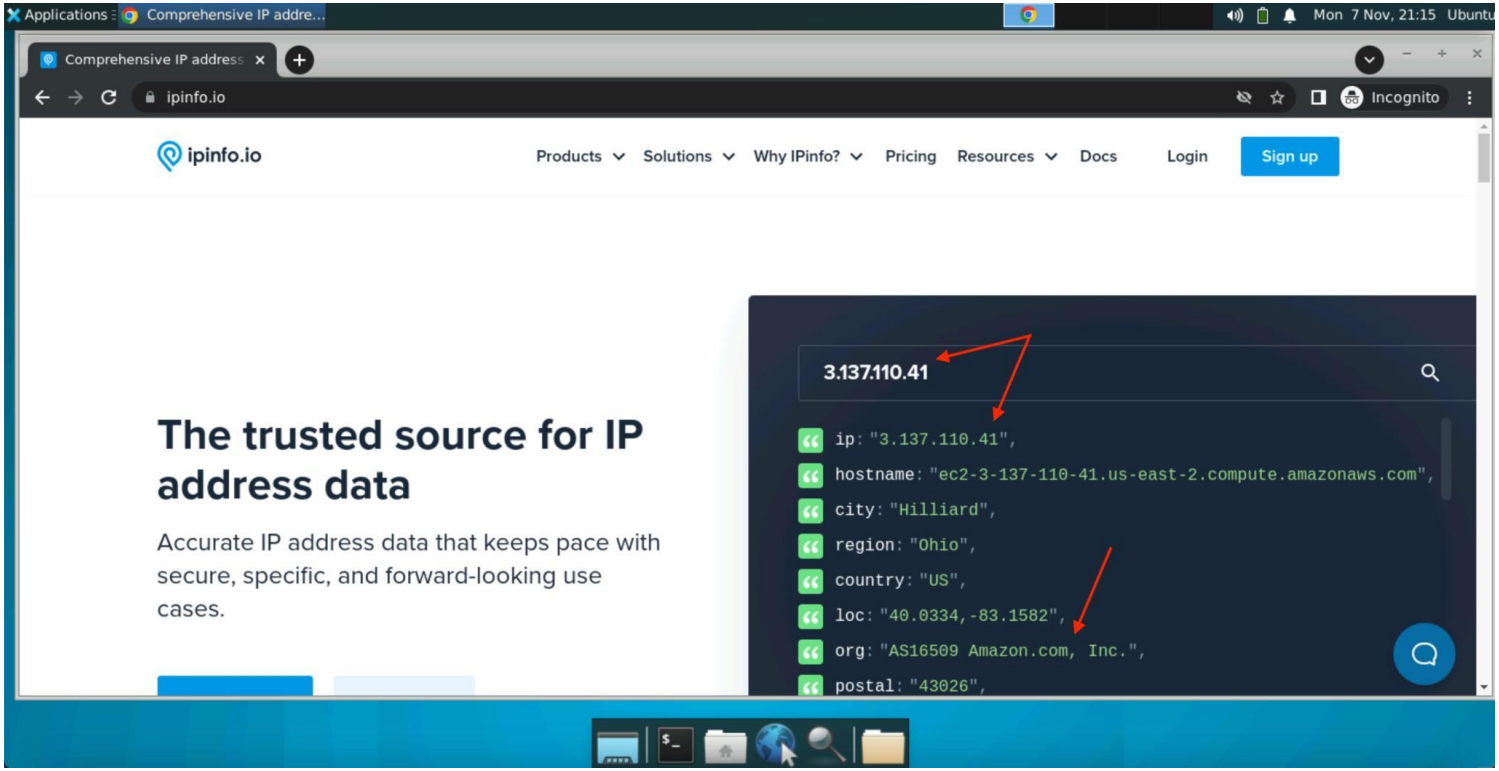
By default, for ZIA customers, a rule will be automatically created for you to send all traffic to ZIA. The Zscaler Private Access (ZPA) option allows traffic that matches the criteria defined to be forwarded to the ZPA cloud. Cloud Connector automatically downloads ZPA Application Segments from your 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.

4. Click **Save**, then **Activate** your changes.

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

5. Return to your **Region1** Workload console and open a new browser window (we recommend that this be an Incognito window to avoid browser caching issues).
6. Navigate to **https://www.ipinfo.io** again and review the results.
7. Note the **IP Address** and **Org** of your workload.
8. Verify that this traffic now exits AWS directly as evidenced by the IP Address and Org. Feel free to experiment with additional Traffic Forwarding Rules.



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.

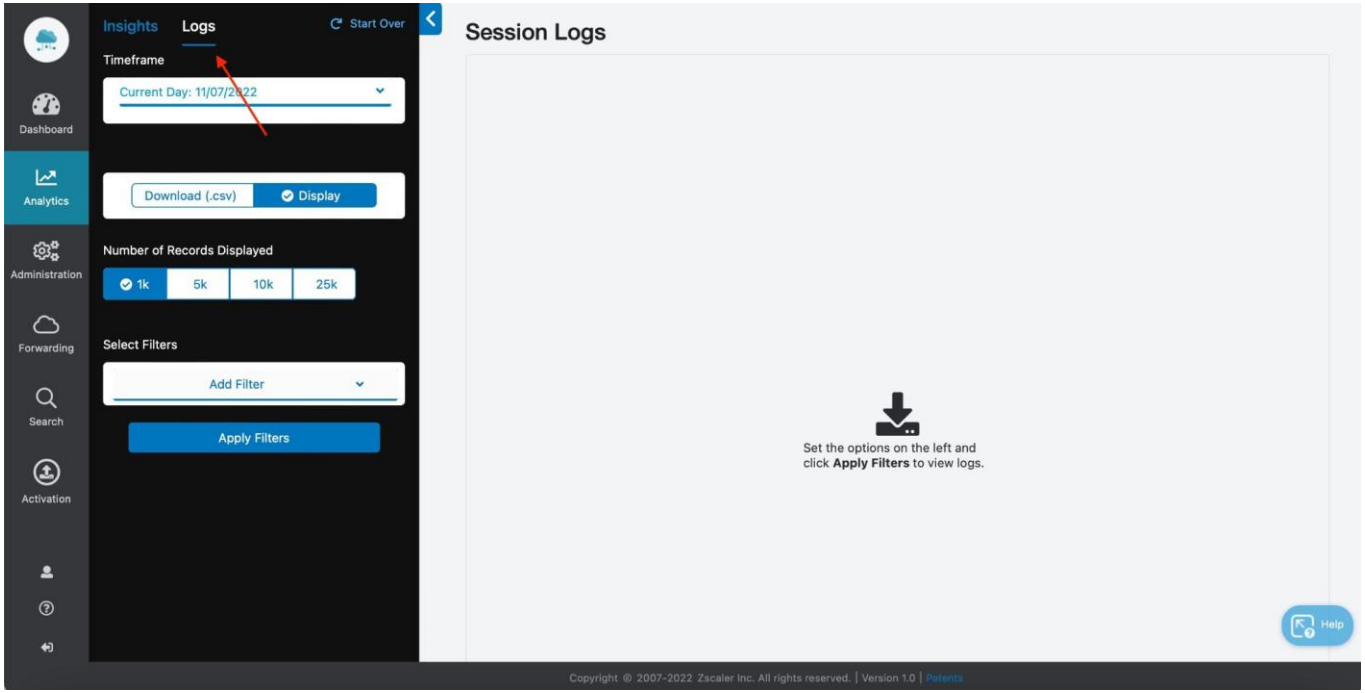
Lab 4: Using Analytics and Logging

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.

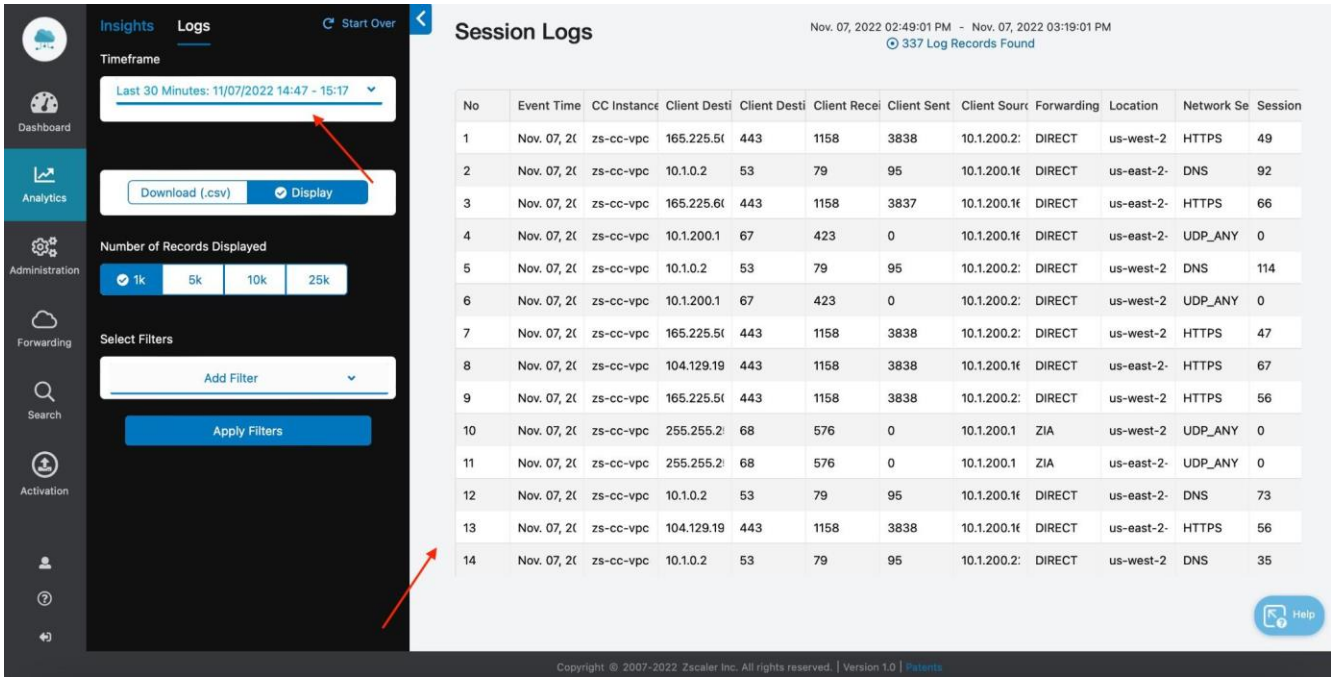
Task 1: Review Insights Logs

In this task, you will access and filter logs to review details for each transaction processed by Zscaler

1. In the Cloud Connector portal, select **Analytics > Session Insights**.
2. Click the **Logs** tab in the upper left.

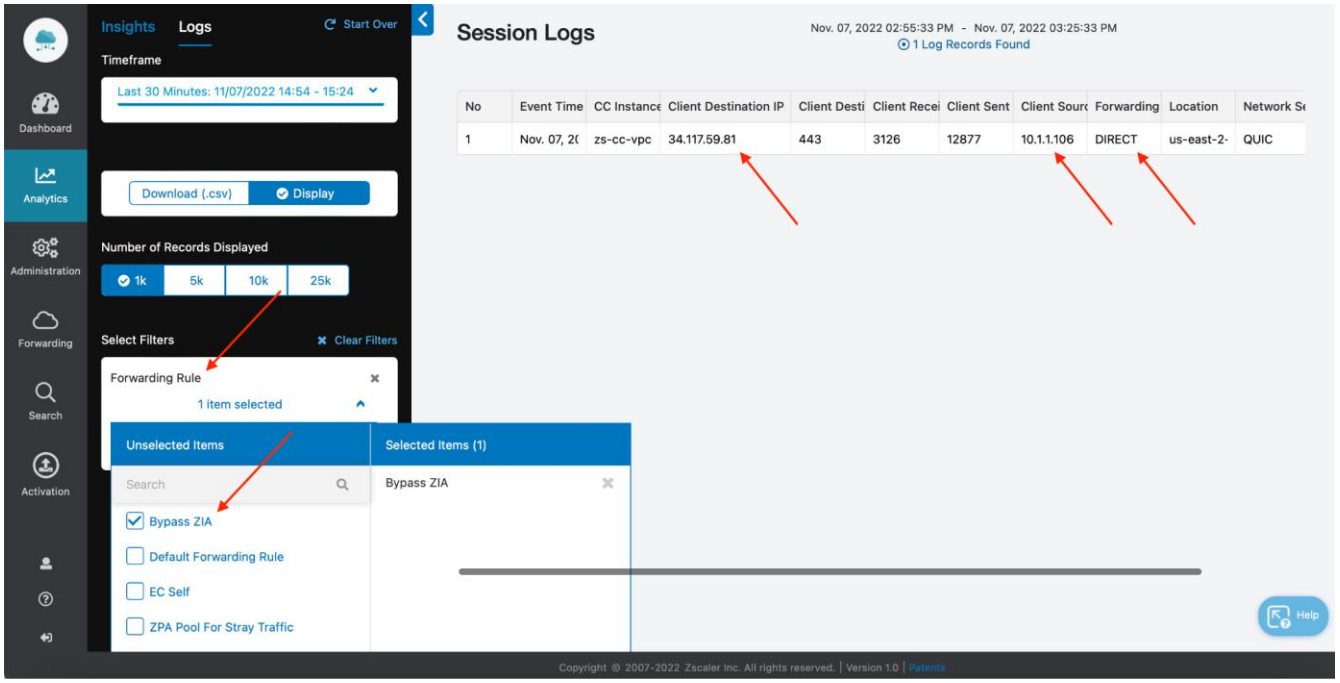


3. Change the Timeframe to **Last 30 Minutes** and click **Apply Filters**.



4. Review the log details for a transaction, including the 5-tuple information on flows seen by the Cloud Connector as well as the disposition of the traffic (where it was sent).
5. Experiment with the available filtering options. For example, try to find the traffic to **https://www.ipinfo.io** you sent earlier.

Hint: *Try filtering by Forwarding Rule.*



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.

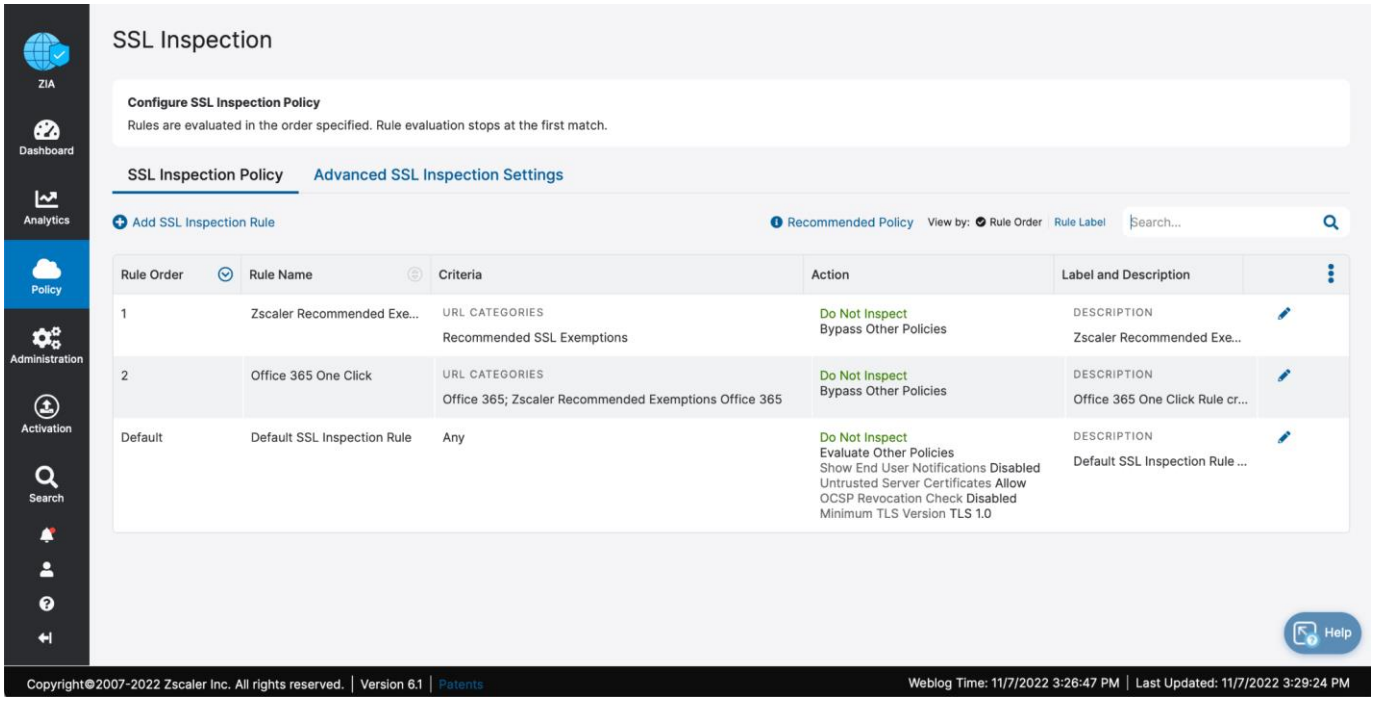
Lab 5: Enforcing Minimum TLS Versions

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. With known vulnerabilities in earlier versions of TLS—namely TLS 1.0 and 1.1—these protocol versions should no longer be used.

Task 1: Configure SSL Inspection Policy

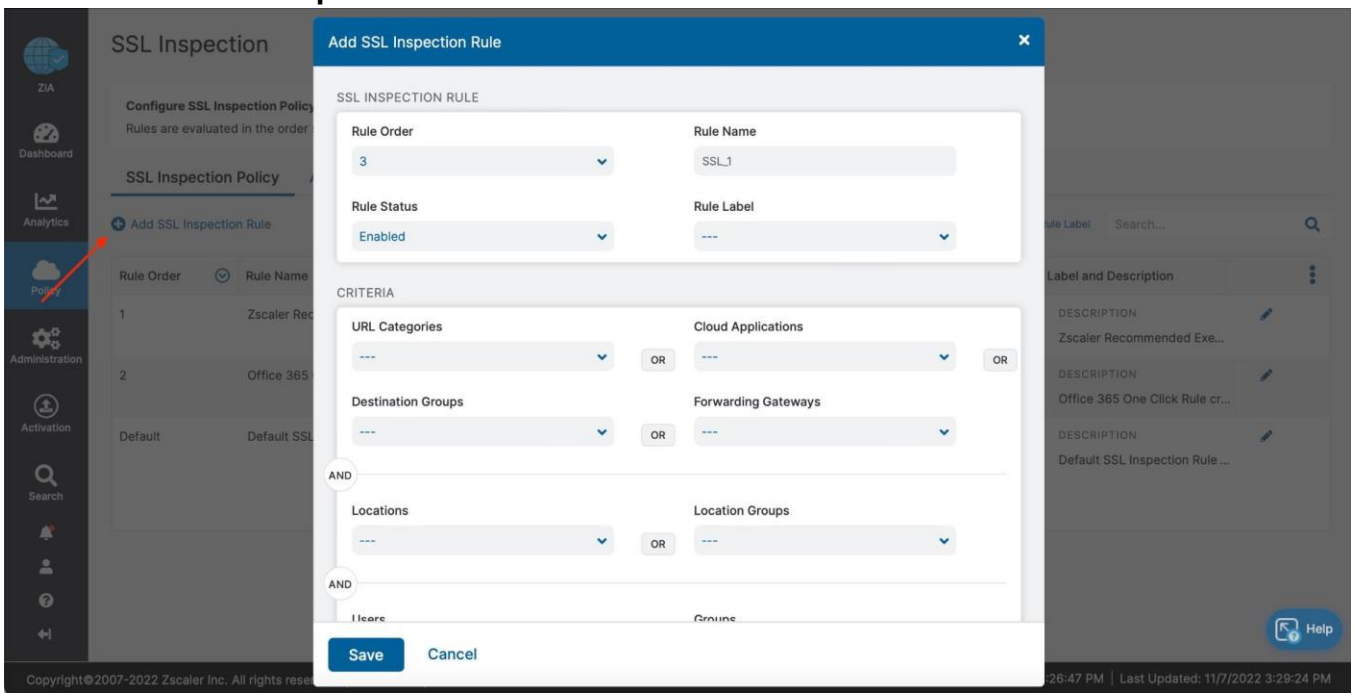
In this task, you will configure a ZIA SSL Inspection policy to enforce the minimum TLS version as 1.2.

1. Navigate to the **ZIA Admin Portal URL** and log in using the credentials provided previously.
2. Select **Policy > ACCESS CONTROL > SSL Inspection**.



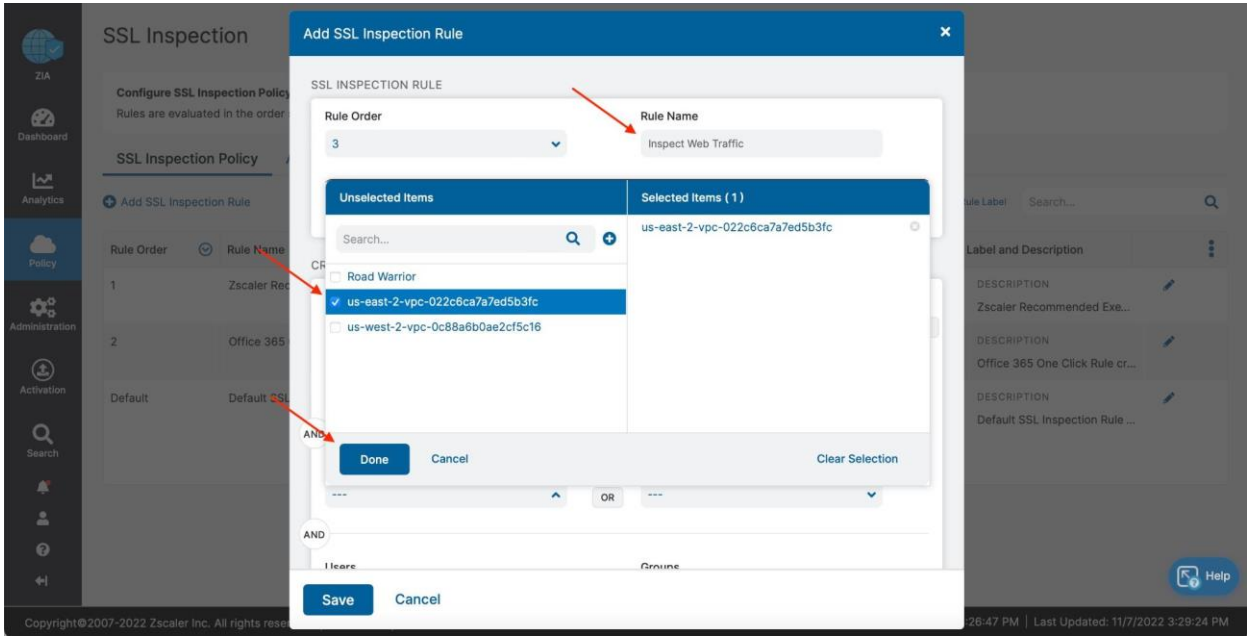
Note: *As you can see from the rules listed, the default behavior is to not inspect SSL traffic. However, you will now create a rule to change that.*

3. Click **Add SSL Inspection Rule**.



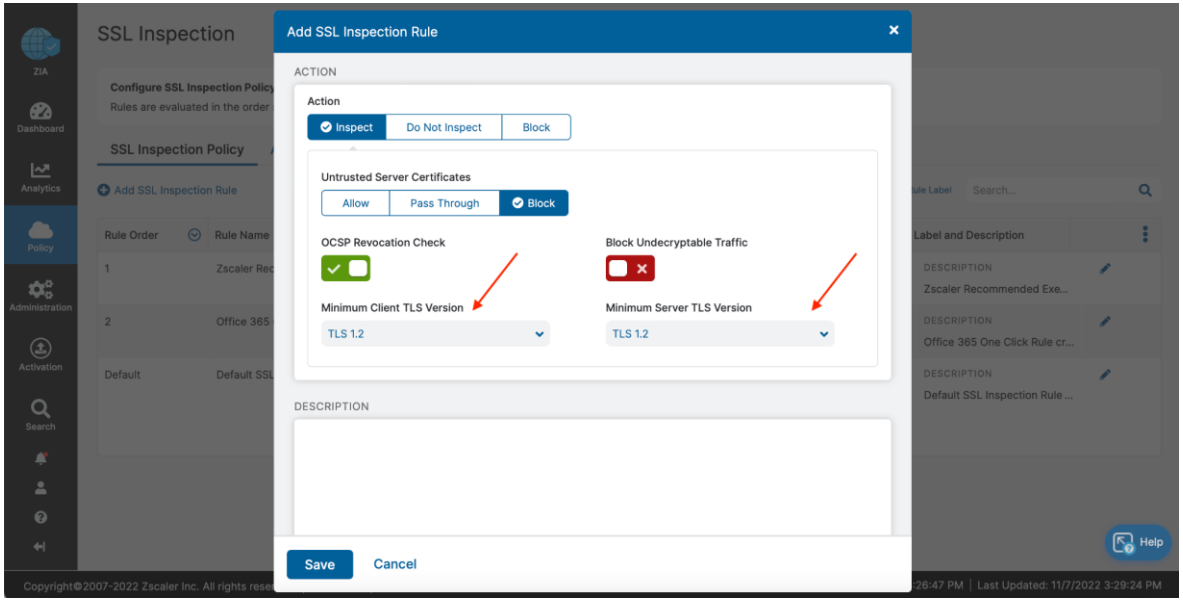
4. Configure the rule as follows:
   a. Enter a **Name**.
   b. Under the Criteria section, click on **Locations** and select your **Region1** and **Region2** Locations.

Note: *The Locations listed take on the naming convention of* ***Region + VPC ID****. Selecting only Region1 would ensure that this policy only applies to your Region1 workload. Likewise, 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.*

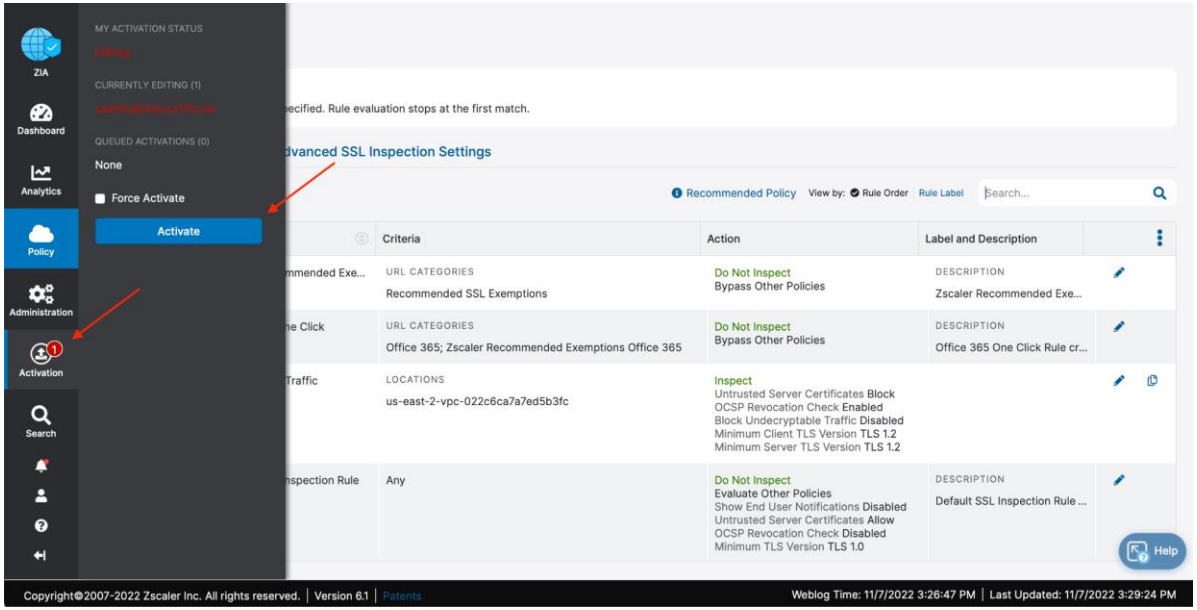


c. Click **Done**.

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



5. Click **Save**.
6. To activate the changes, mouse over the **Activation** menu and click **Activate**.



Task 2: Verify SSL Inspection Policy

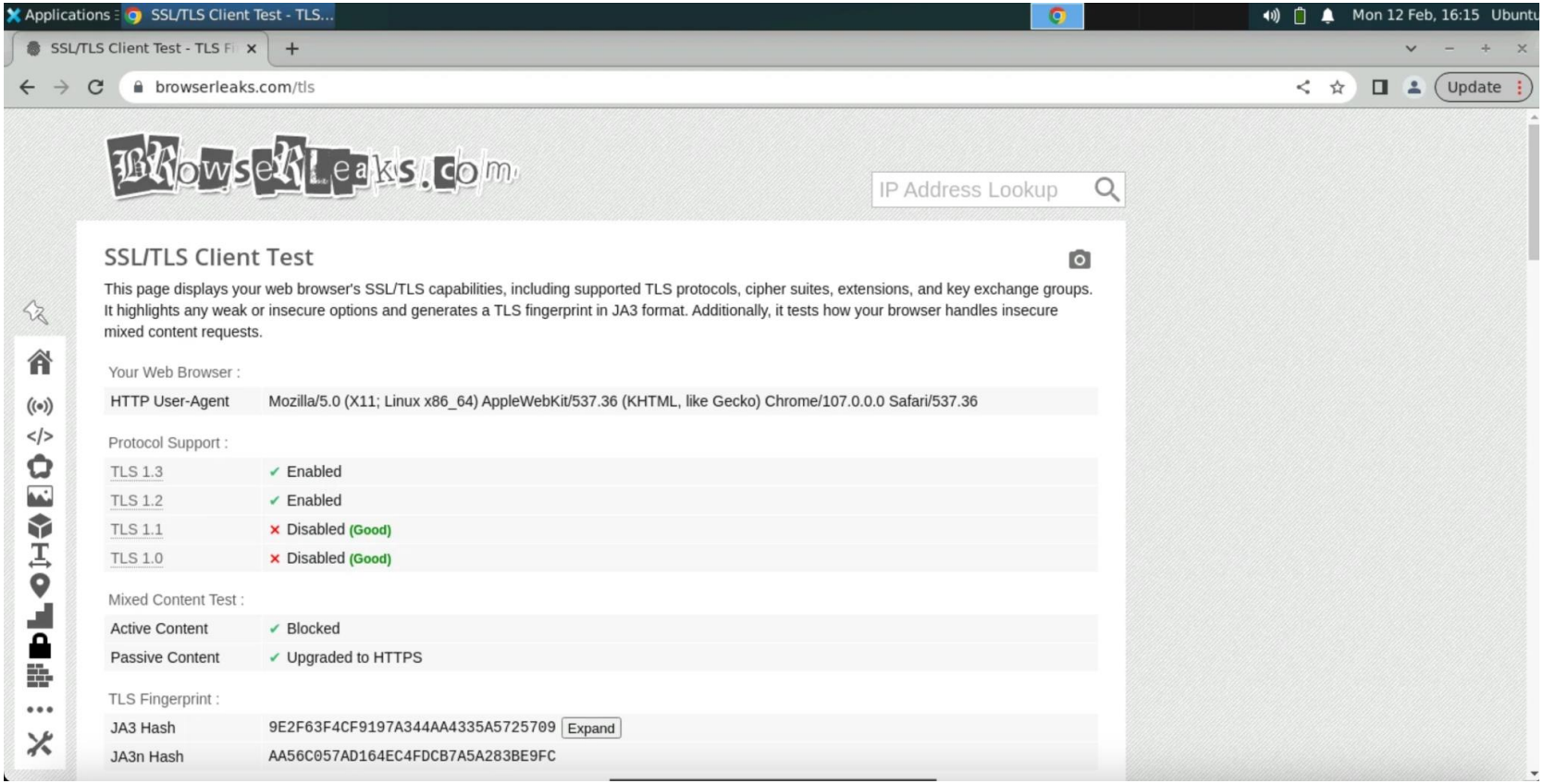
In this task, you will verify that the configured SSL Inspection Policy has the desired effect.

1. Return to the **Region1** Workload, open a web browser and navigate to https://browserleaks.com/ssl .

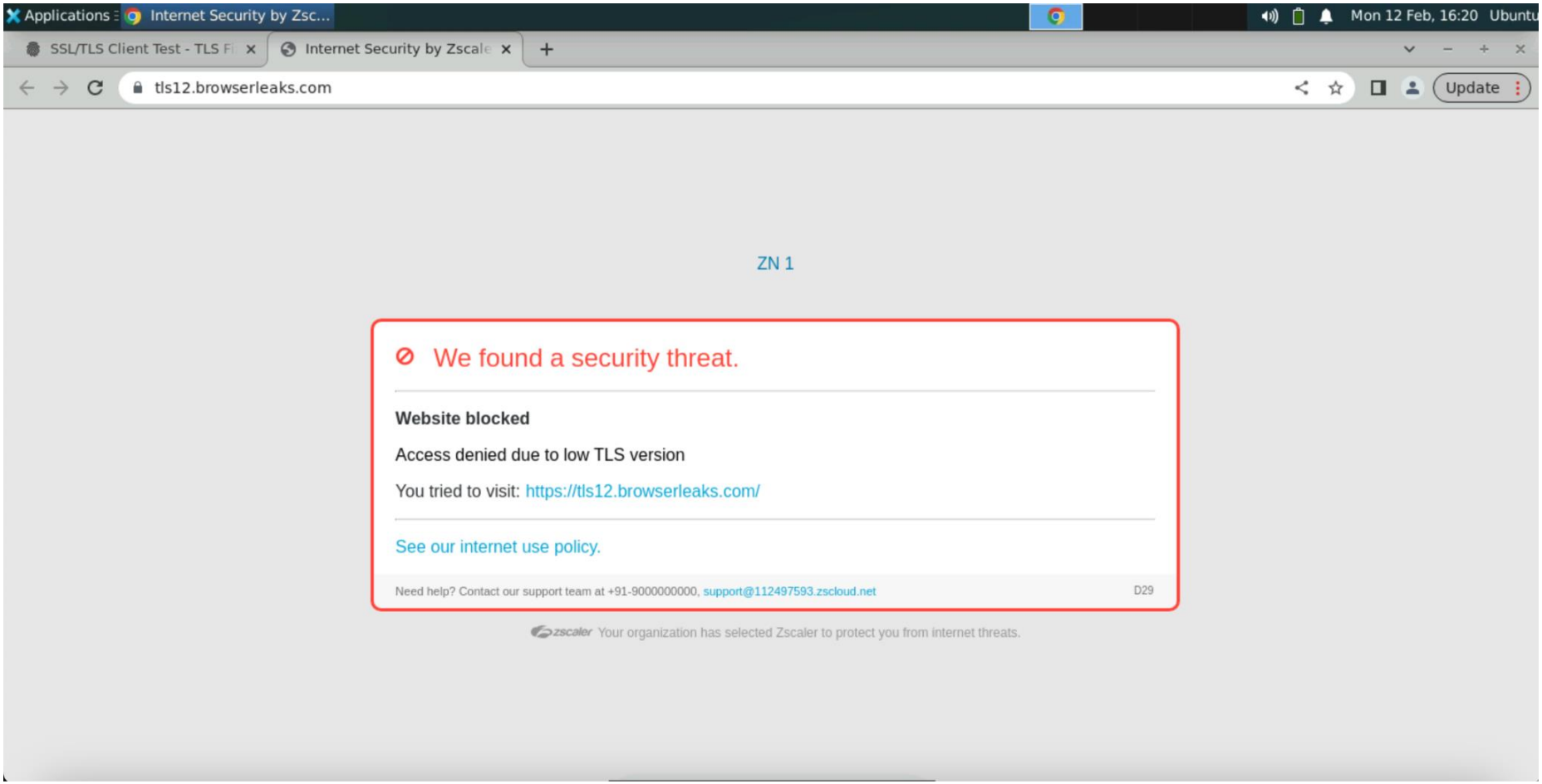
Note: *The site will automatically begin to test the TLS versions enabled and accessible on your browser.*

---

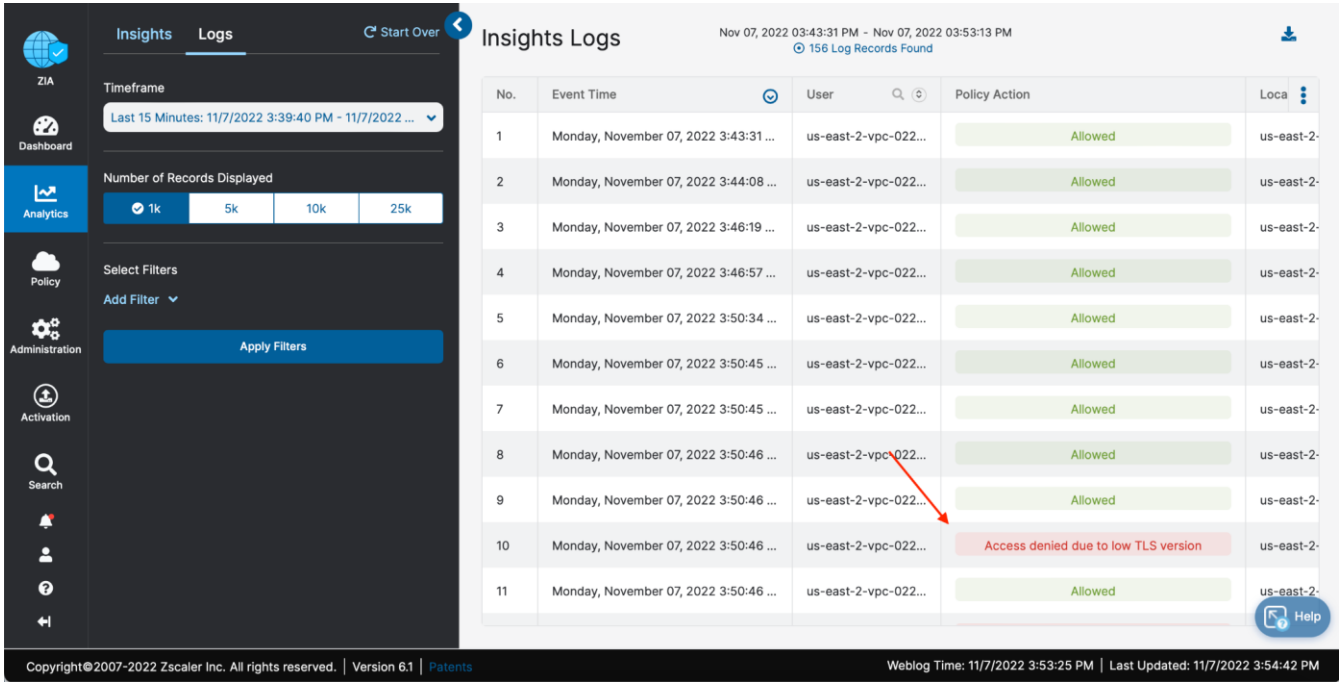
2. On the page that appears, note the Protocol Support section of the page. Clicking on the links for TLS 1.2 and TLS 1.3 will open a new page that negotiates TLS 1.2 or TLS 1.3, respectively. .



3. Return to the ZIA Admin portal and select **Policy > ACCESS CONTROL > SSL Inspection**.
4. Click the “pencil” icon next to your SSL policy to edit it.
5. Scroll to the Action section and, under **Minimum Client TLS Version** and **Minimum Server TLS Version**, select **TLS 1.3**.
6. Click **Save**.
7. Activate the changes.
8. Return to the **Region1** Workload, and refresh the web browser .
9. You should notice that clicking on the TLS v1.2 (or below) links now result in a **blocked page**. This is because the Zero Trust Exchange is enforcing TLS 1.3 in the environment.



10. To verify this behavior, return to the ZIA Admin portal and select **Analytics > Web Insights**.
11. Select the **Logs** tab (upper left).
12. Narrow the Timeframe to **Last 15 Minutes** and click **Apply Filters**.

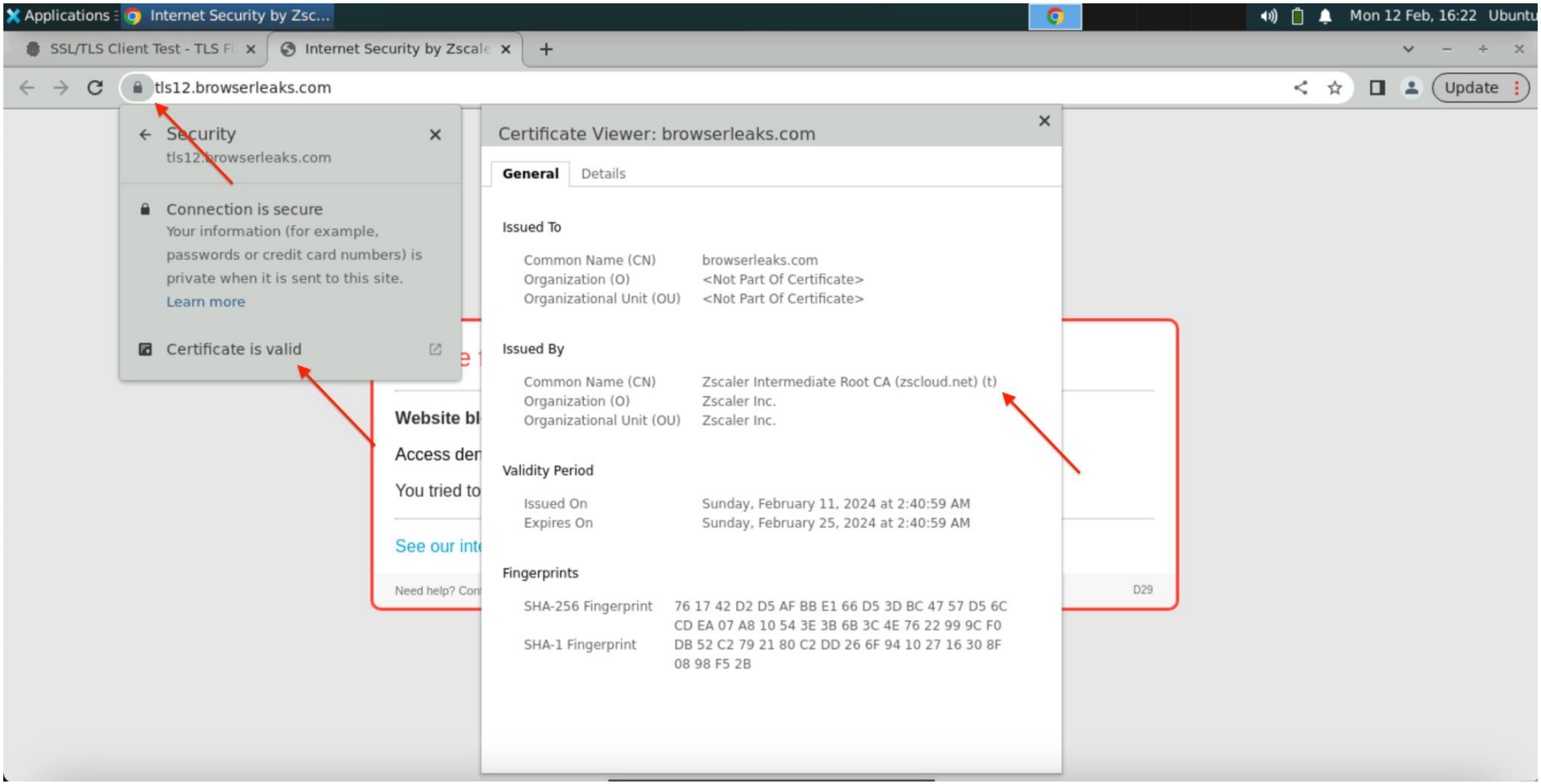


13. Review the log entries and focus on the policy block due to TLS version.

Note: *Zscaler proxies SSL connections by inserting itself into the HTTPS exchange. The Cloud Connector appliance does not inspect this traffic.*

---

14. Return to the **Region1** Workload, and in the web browser click the “lock” icon in the upper left (next to the address bar).
15. Click the **Connection is secure** link, followed by **Certificate is valid**.
16. Verify 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.

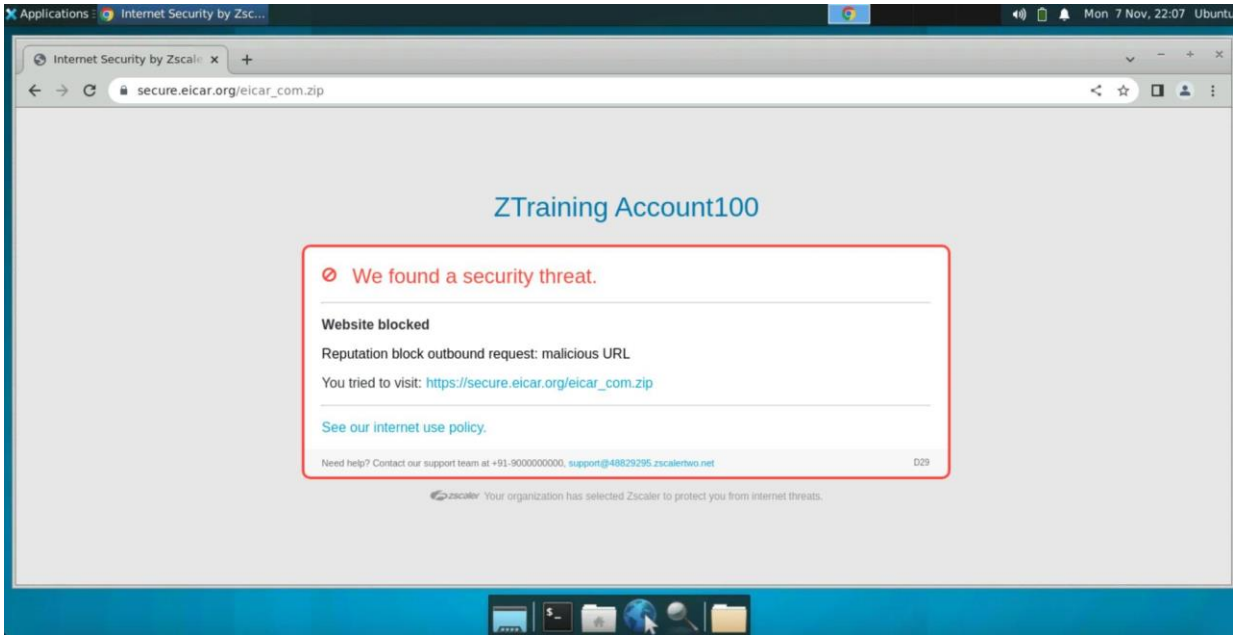
Lab 6: Protecting Against Malicious Payloads, Phishing, and BotNet

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. Apart from viruses, Zscaler can also detect other threats such as phishing sites and BotNets.

Task 1: Test Zscaler Security Protections

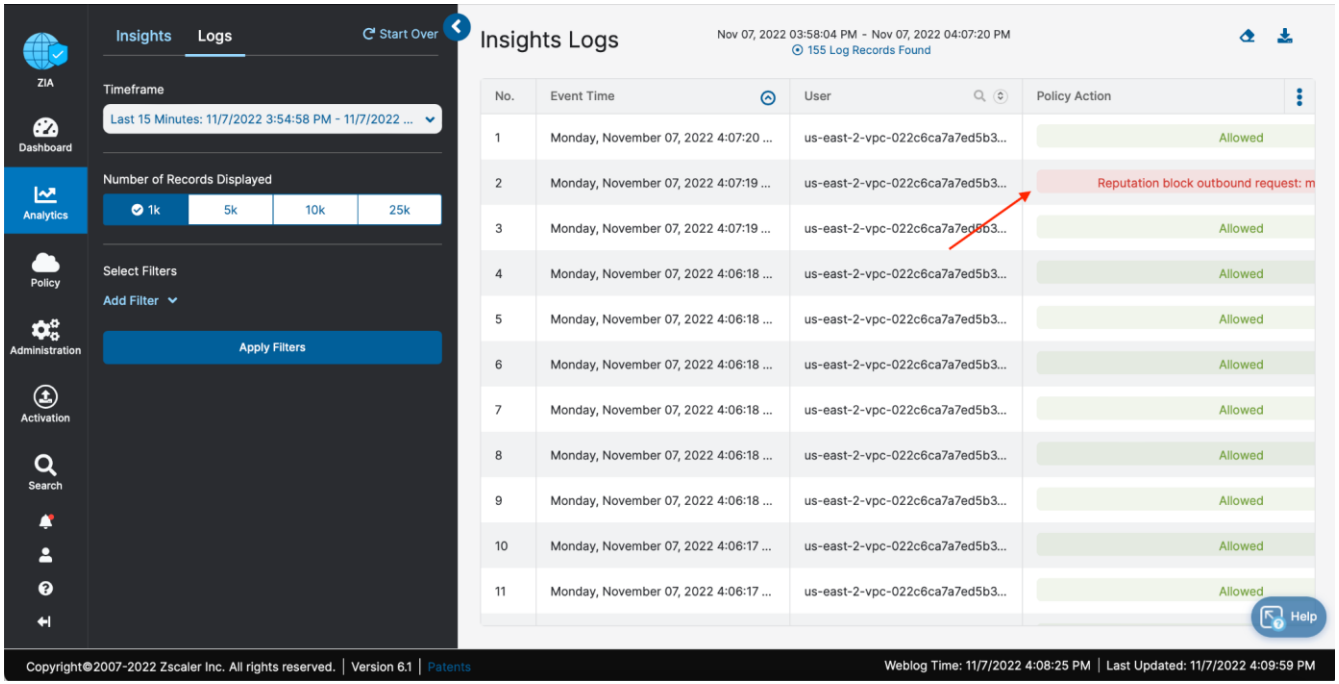
In this task, you will use a well-known test file from EICAR (European Institute for Computer Anti-Virus Research) to trigger ZIA security protections.

1. From the **Region1** Workload, browse to **https://www.eicar.org/download-anti-malware-testfile**.
2. On the right side of the page, click one of the test files to download (eicar.com, eicar.com.txt, eicar_com.zip, etc.).
3. Verify that Zscaler Internet Access automatically blocks this download attempt.



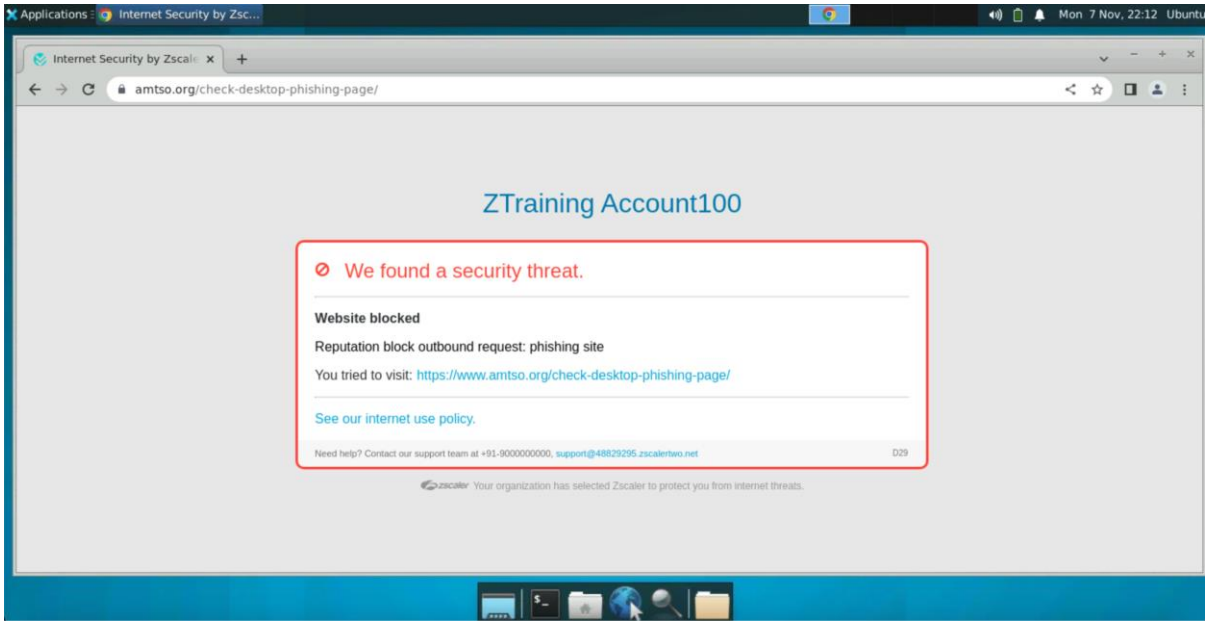
Note: *This is because the file contained malicious code. Even malicious files hidden inside compressed archives (ZIP) can be blocked.*

4. Return to the ZIA Admin portal and select **Analytics > Web Insights**.
5. Select the **Logs** tab (upper left).
6. View the log entry for the file you just attempted to download.

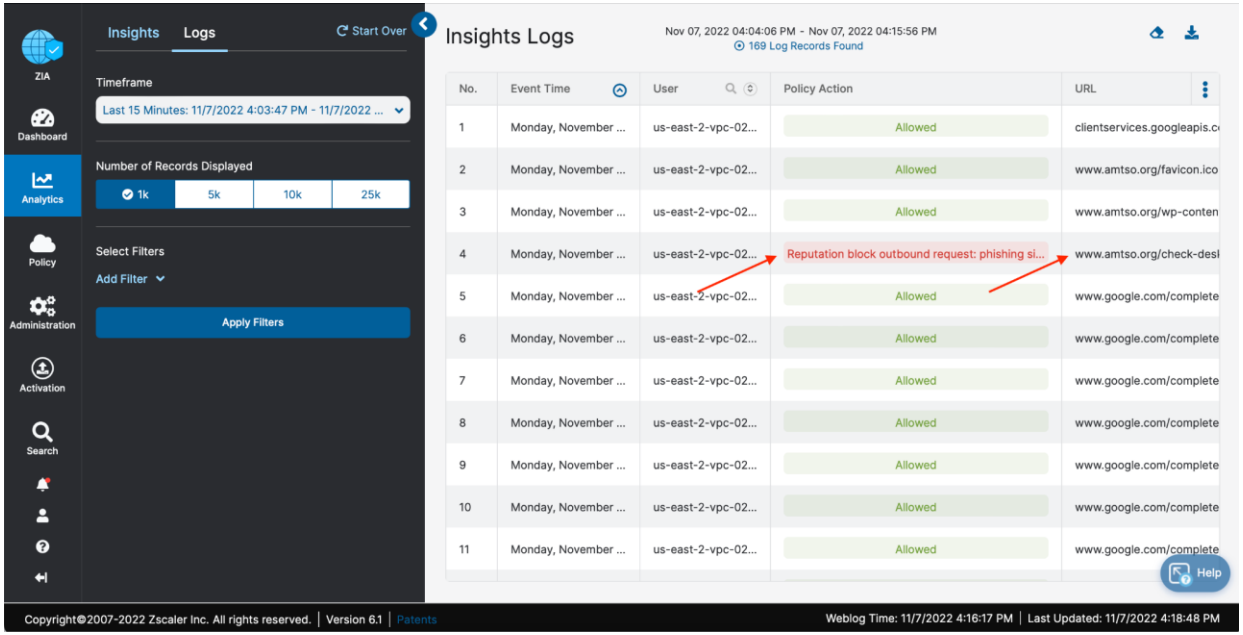


Note: *Be patient, it may take 2-3 minutes for logging to catch up.*

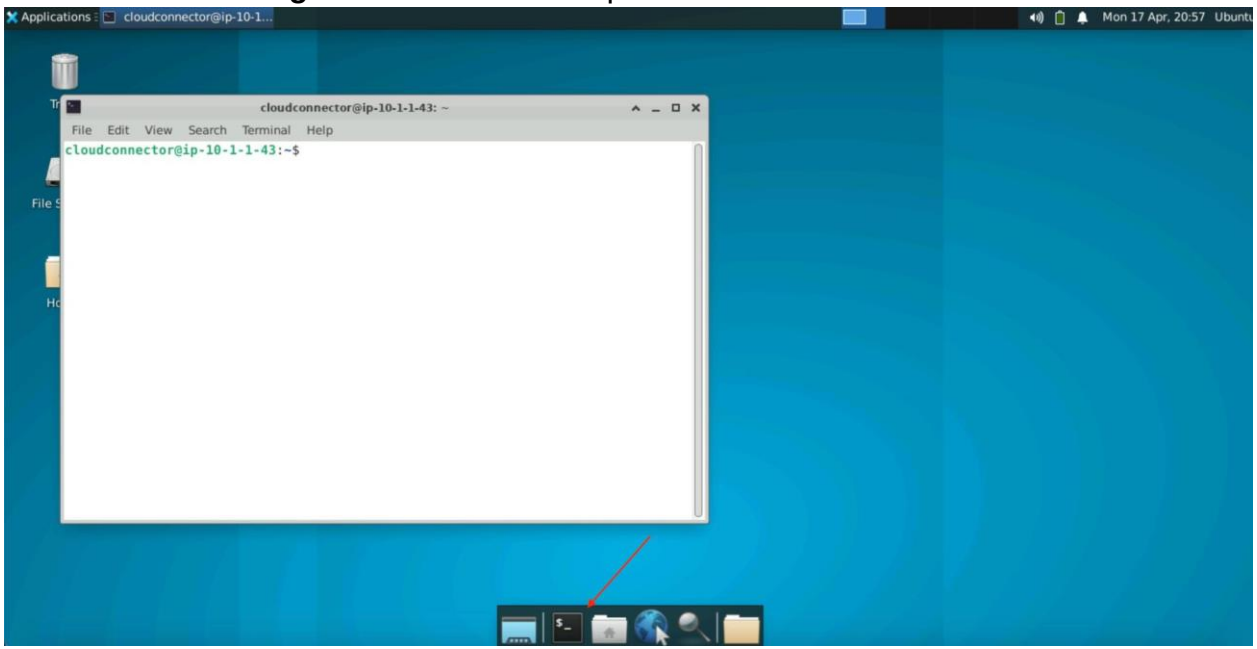
7. From your **Region1** Workload, navigate to **https://www.amtso.org/check-desktop-phishing-page**.



8. Verify that the threat was blocked since this is a known phishing site.
9. As before, on the ZIA Admin Portal **Analytics** (Log tab) page, review the log entry for the site you just attempted to visit.



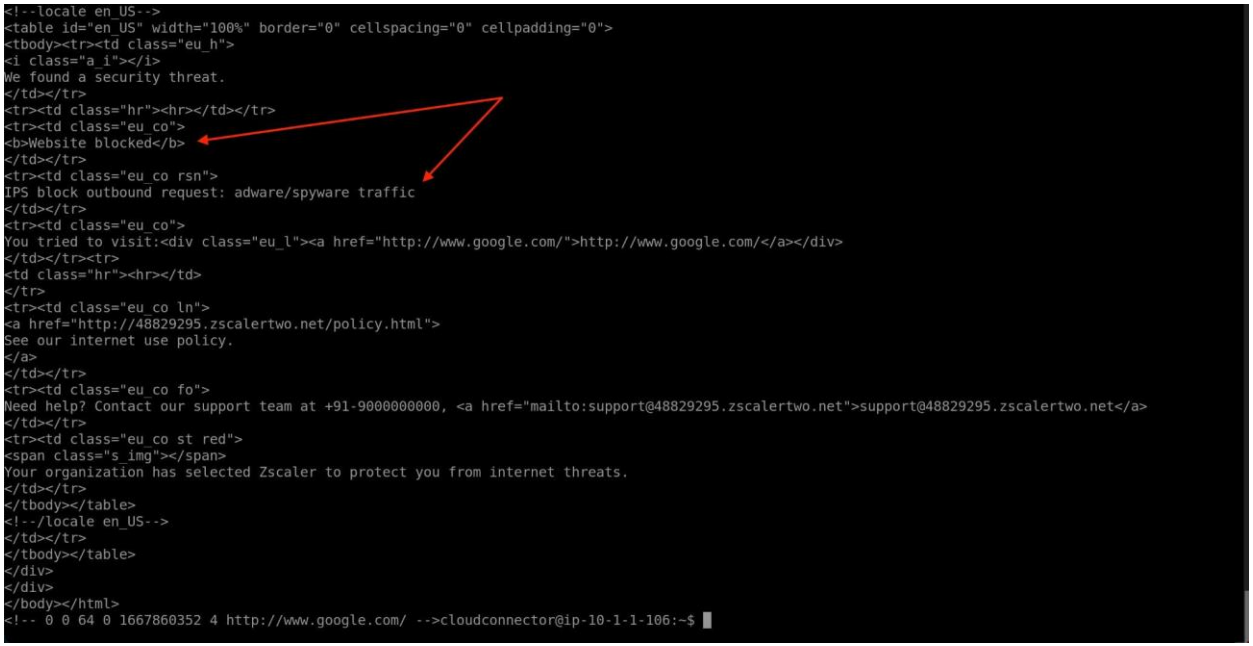
10. Return to the **Region1** Workload and open a terminal.



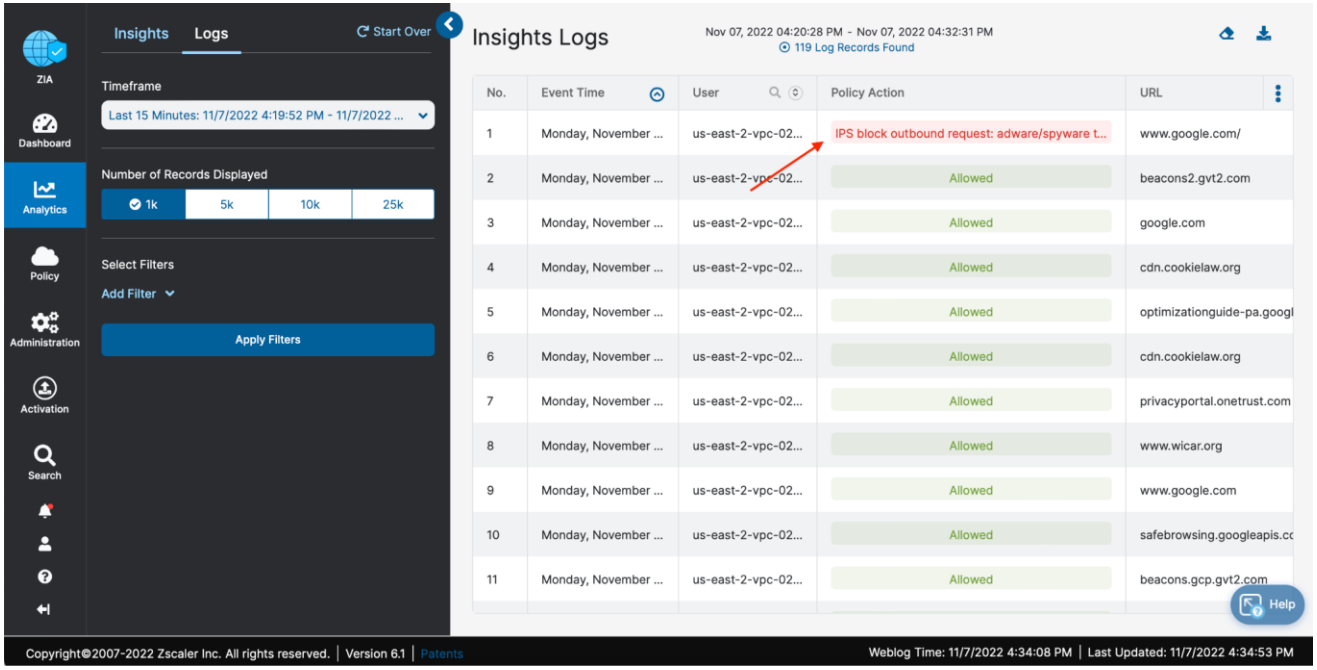
11. At the Command Line, enter the following command: **curl -A “BlackSun” www.google.com**.

> ***Note:*** *This is a test that spoofs the BlackSun browser agent and will trigger an IPS response from most firewalls.*

12. Verify this threat too is blocked by ZIA IPS automatically.



13. On the ZIA Admin Portal Analytics (Log tab) page, review the log entry for this IPS Block.



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.

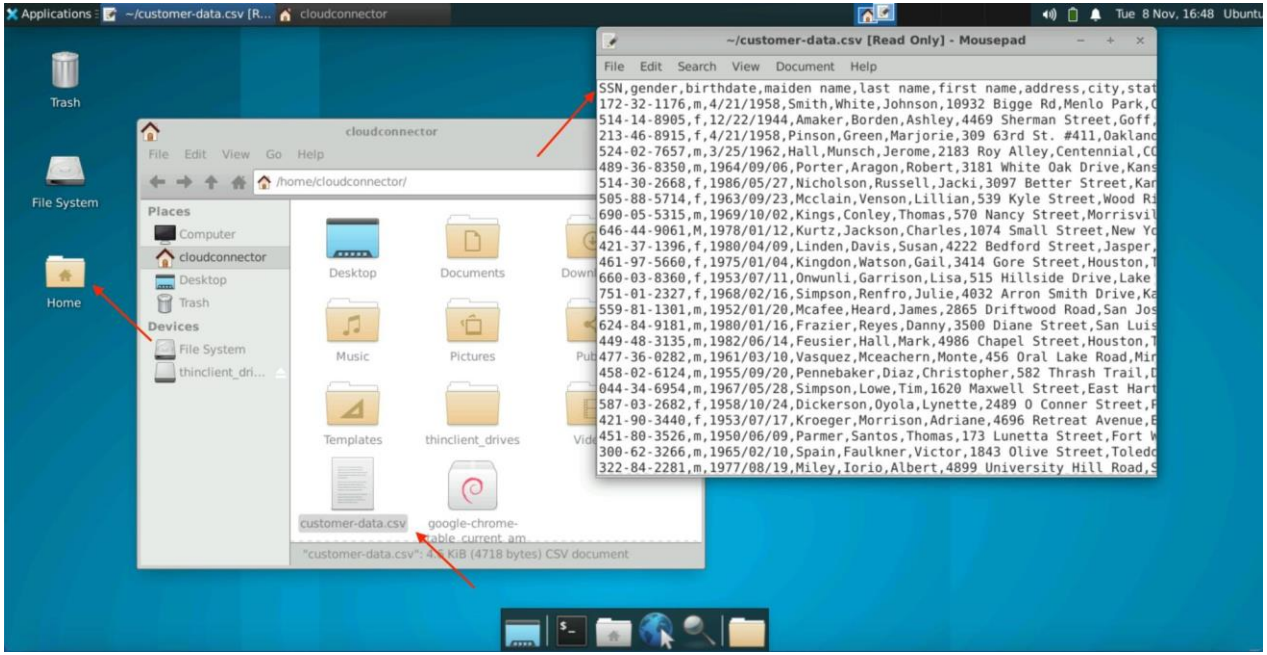
Lab 7: Enforcing a Data Loss Prevention Policy

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 lab, we demonstrate how Zscaler prevents data exfiltration.

Task 1: Configure DLP Policy

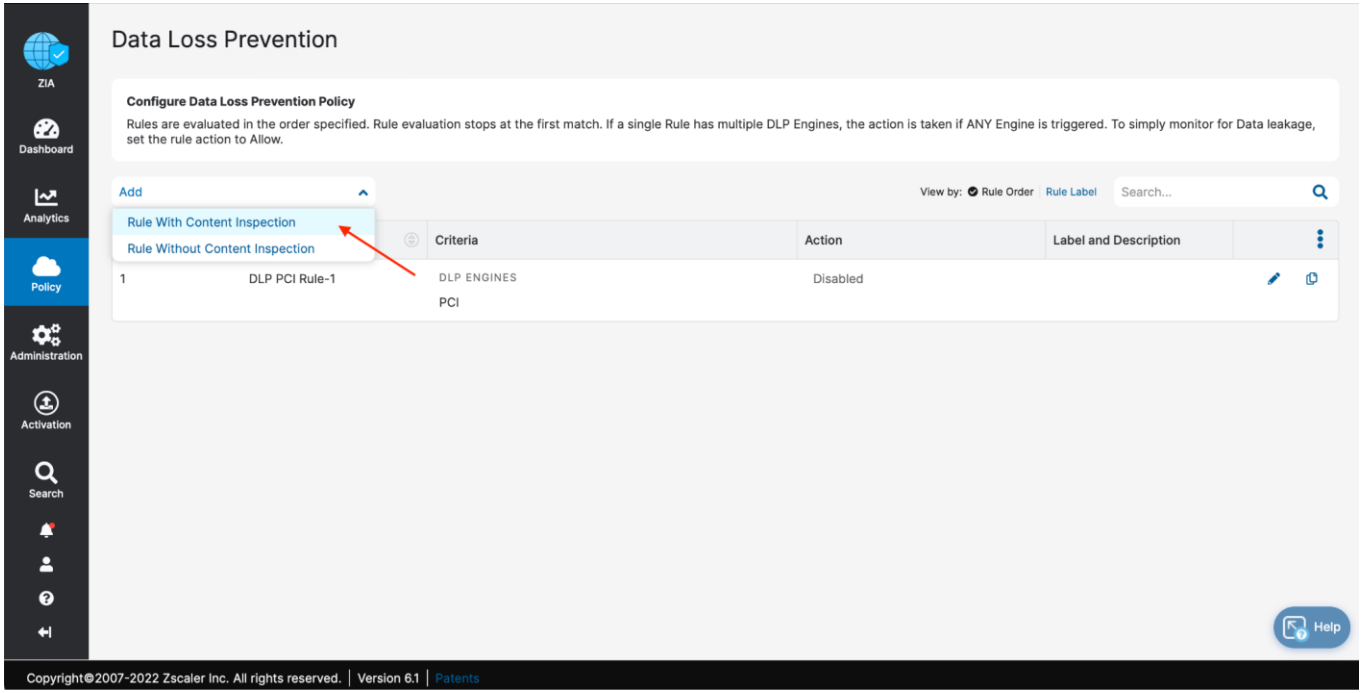
In this task, you will configure a DLP Policy to block the exfiltration of Personal Identifiable Information (PII).

1. From the **Region1** Workload, take a look at the **customer-data.csv** file located in the **Home** folder on the Desktop by double-clicking it.



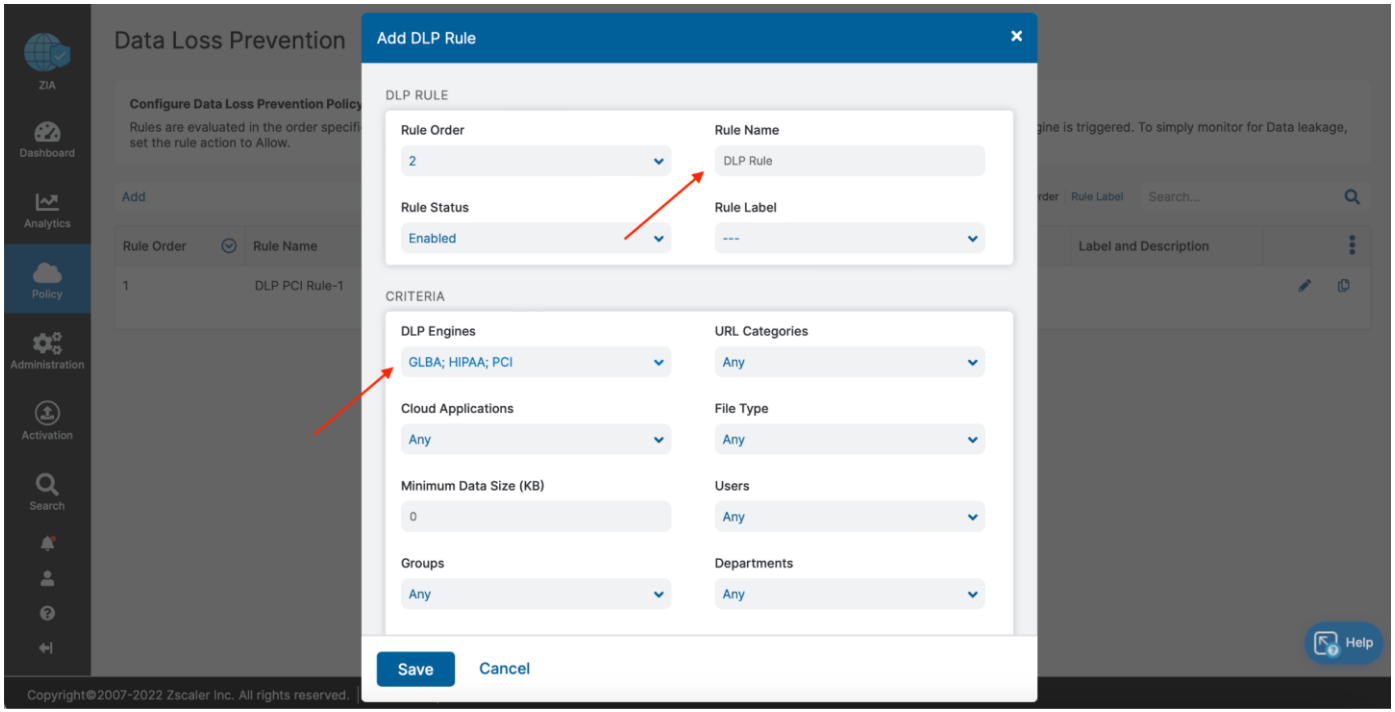
Note: *Ensure you review the CSV file and not the TXT file. This file contains fake Personal Identifiable Information (PII). Let's assume a disgruntled employee tries to exfiltrate this data.*

2. In the ZIA Admin portal, select **Policy > DATA LOSS PREVENTION > Data Loss Prevention**.
3. Click the **Add** button and select **Rule With Content Inspection**.



4. Configure the rule as follows:
   a. Enter a **Name**.

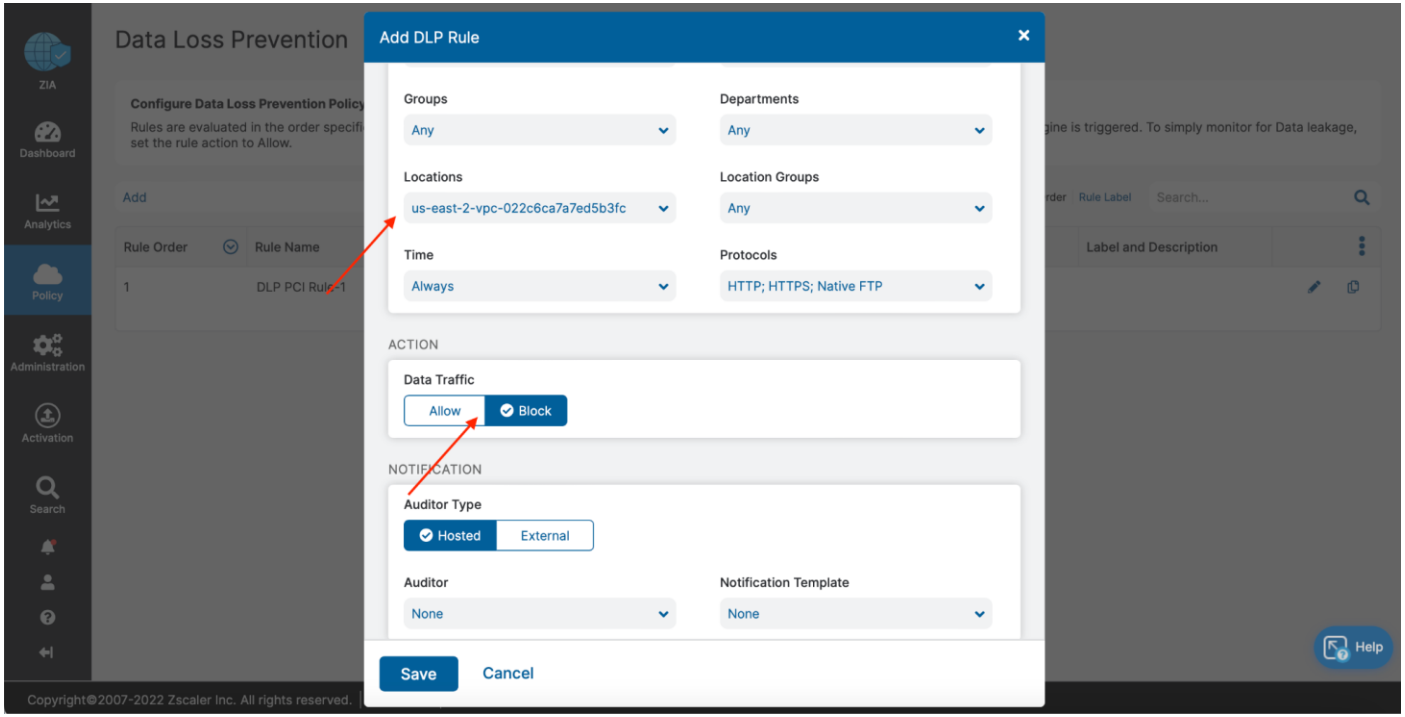
b. Under the Criteria section, click **DLP Engines** and select **GLBA**, **HIPAA**, and **PCI**. Click **Done**.



c. Under the Criteria section, click **Locations** and select your **Cloud Connector Locations** from the dropdown menu.

d. Click **Done**.

e. Under the Action section, select **Block**.

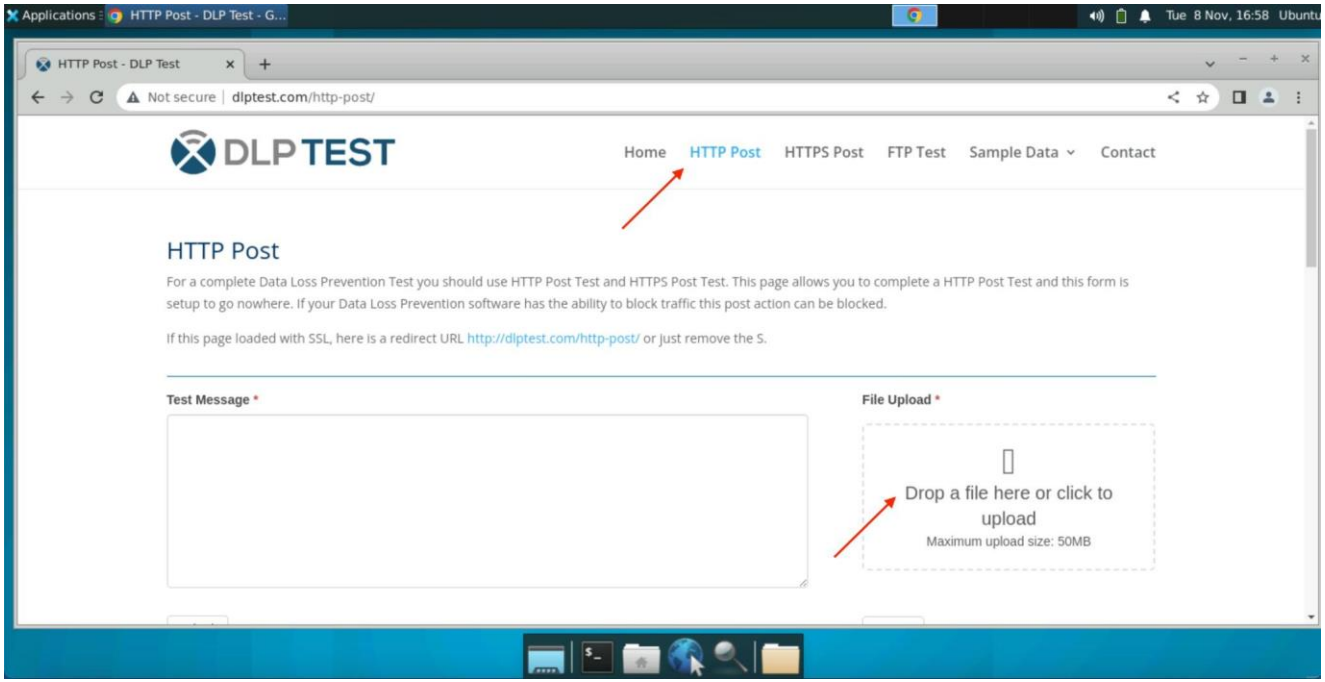


5. Click **Save**.
6. Activate the changes.

Task 2: Verify DLP Policy

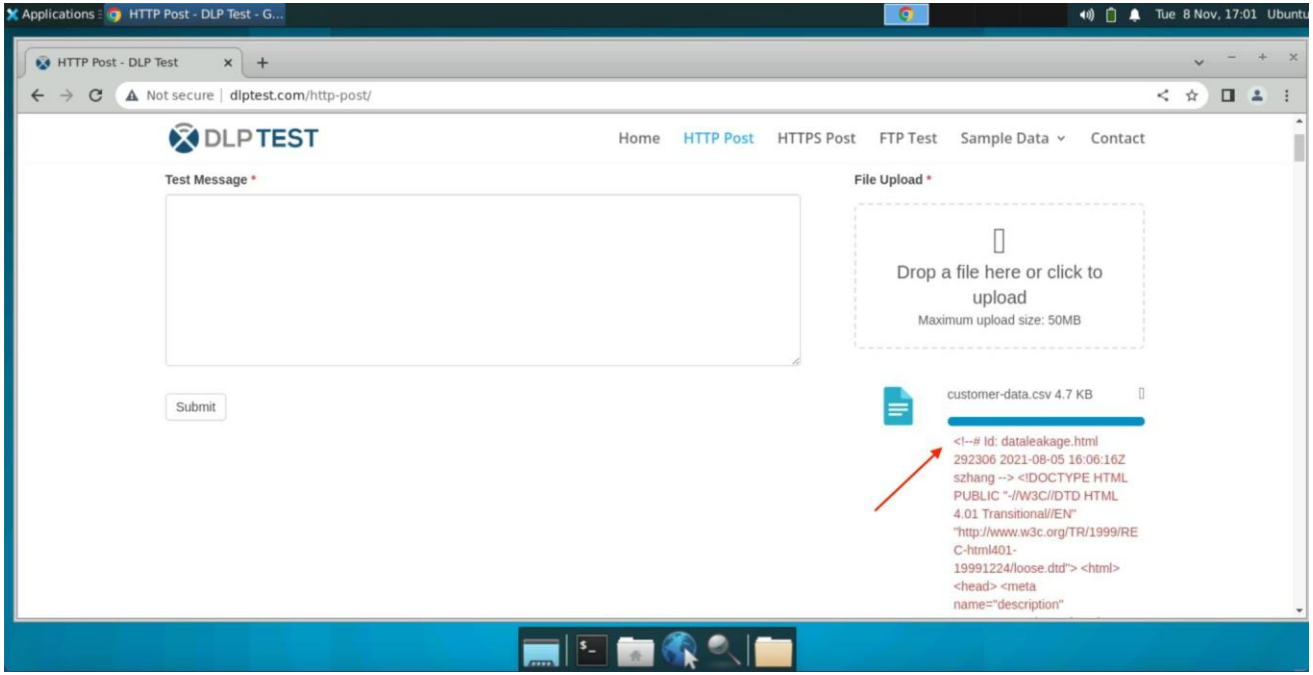
In this task, you will verify that the configured DLP Policy has the desired effect.

1. Return to the **Region1** Workload, open a web browser and navigate to **https://www.dlptest.com**.
2. 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 lab.*

3. Click the button to upload a file on the right and select the **customer-data.csv** file you reviewed earlier.
4. Click **Upload**.

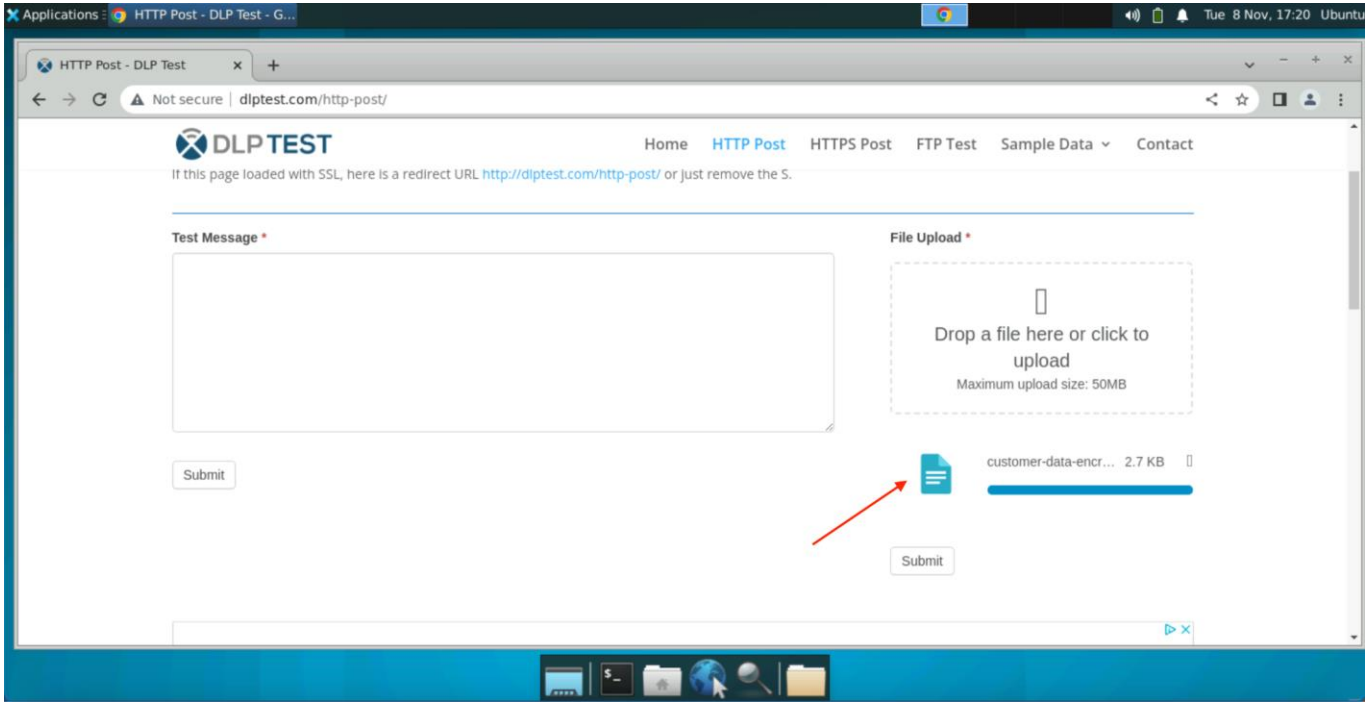


5. Verify that the file is immediately blocked from being uploaded.

Note: *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). This file is the encrypted version of the file you just attempted to upload.*

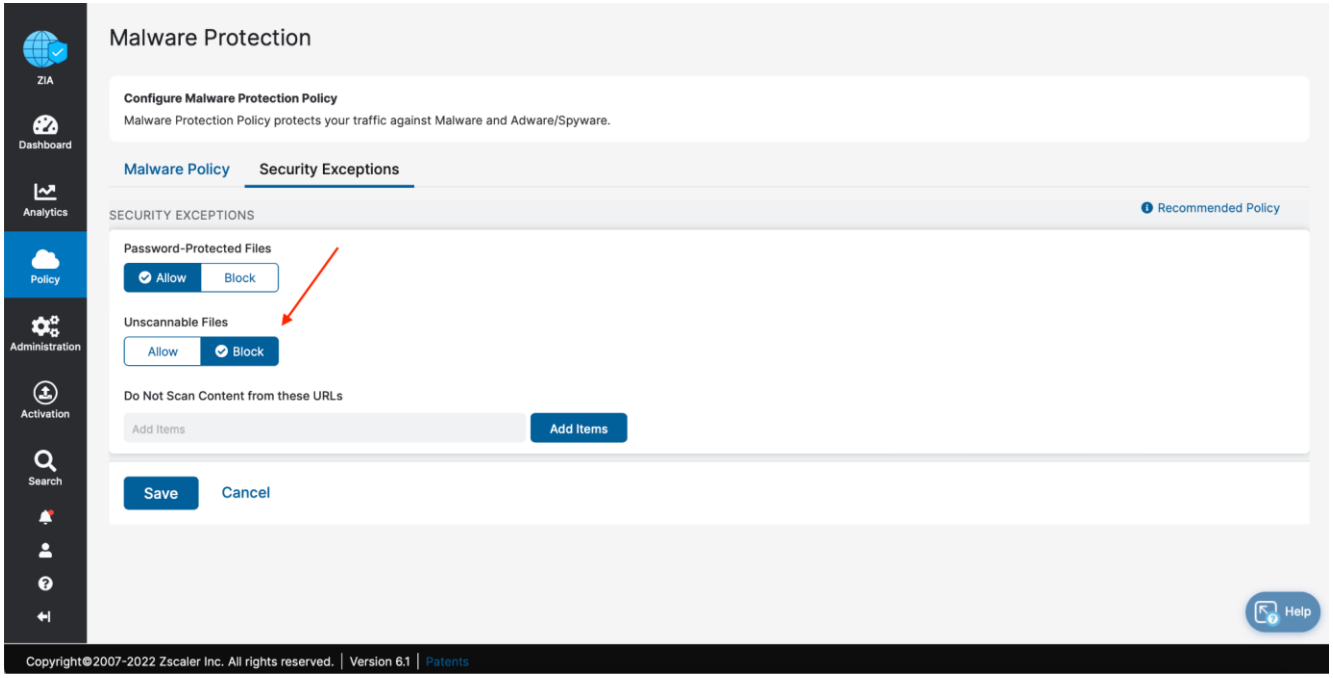
---

6. Repeat the upload test with **customer-data-encrypted.txt**.



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.

7. Return to the ZIA Admin portal and select **Policy > SECURITY > Malware Protection**.
8. Click the **Security Exceptions** tab.
9. For **Unscannable Files**, choose the **Block** option.



10. Click **Save**.
11. Activate the changes.
12. Return to the **Region1** Workload and try to upload the **customer-data-encrypted.txt** file again.
13. Verify that this time, the file failed to upload.

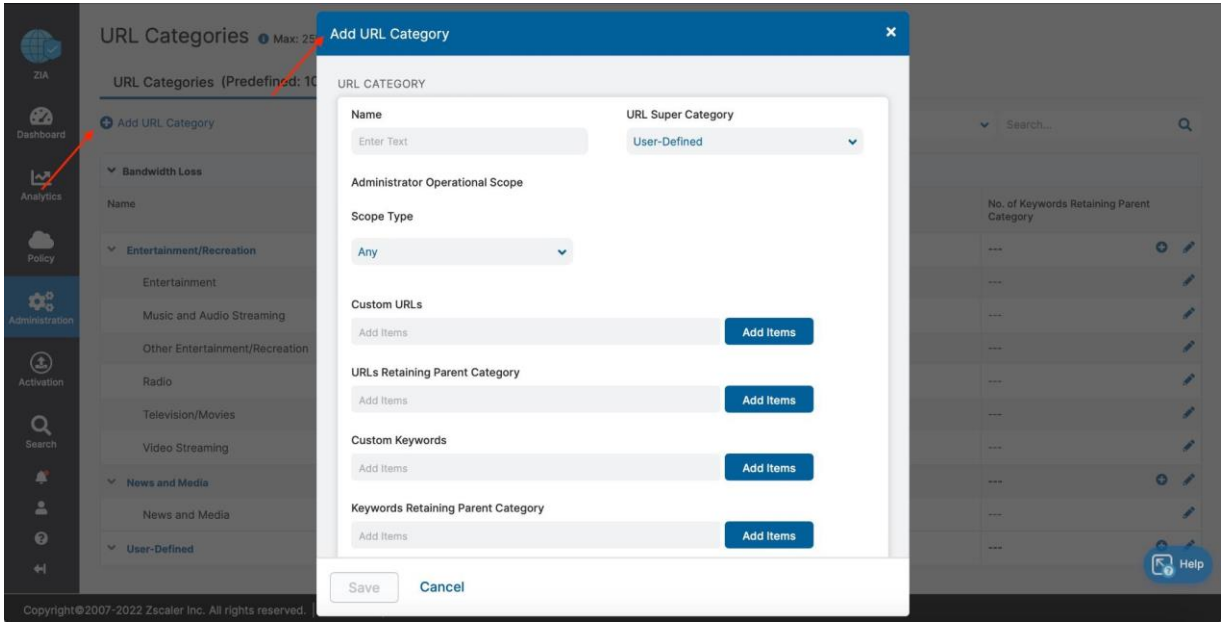
Lab 8: Controlling Access to Specific Resources on Websites

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

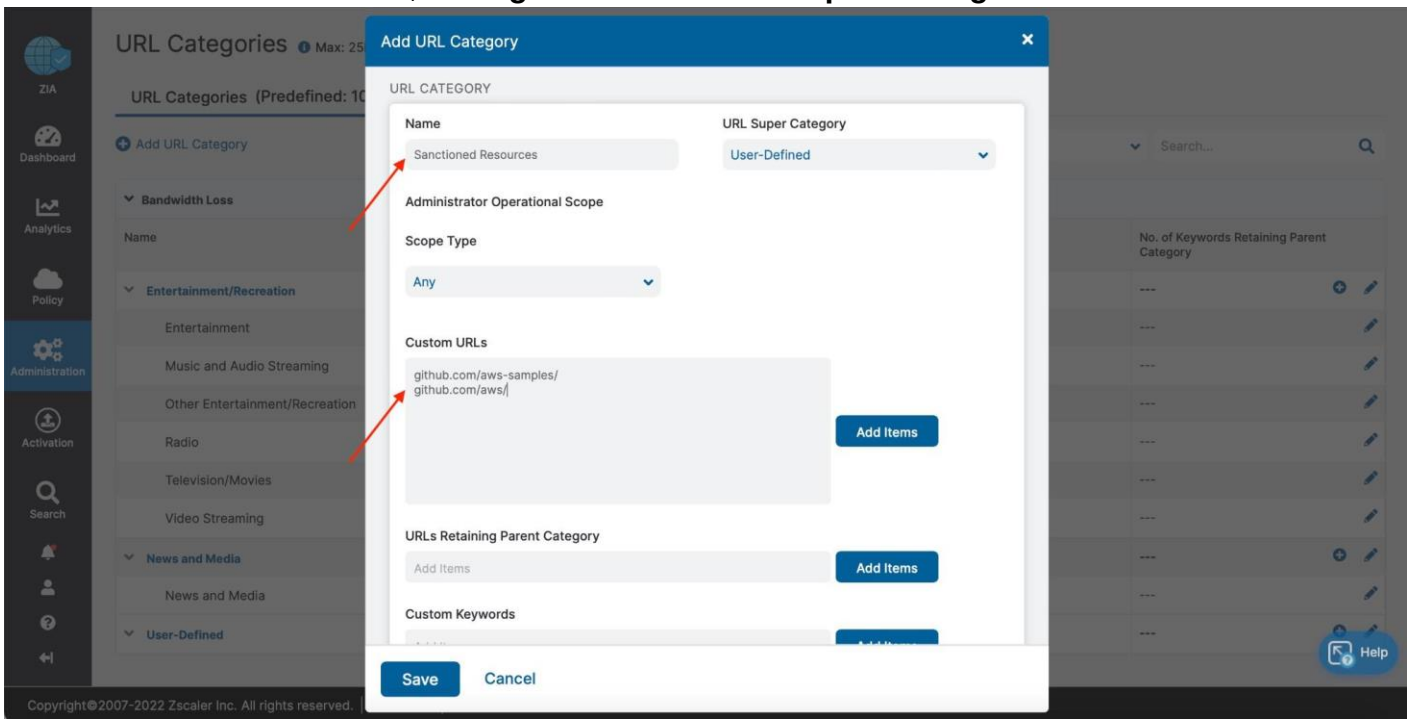
Task 1: Configure URL Filtering Rule

In this task, you will configure custom URL categories which will then be used in URL Filtering Rules to control access to sanctioned/unsanctioned Github repositories.

1. In the ZIA Admin portal, select **Administration > ACCESS CONTROL > URL Categories**.
2. Click **Add URL Category**.

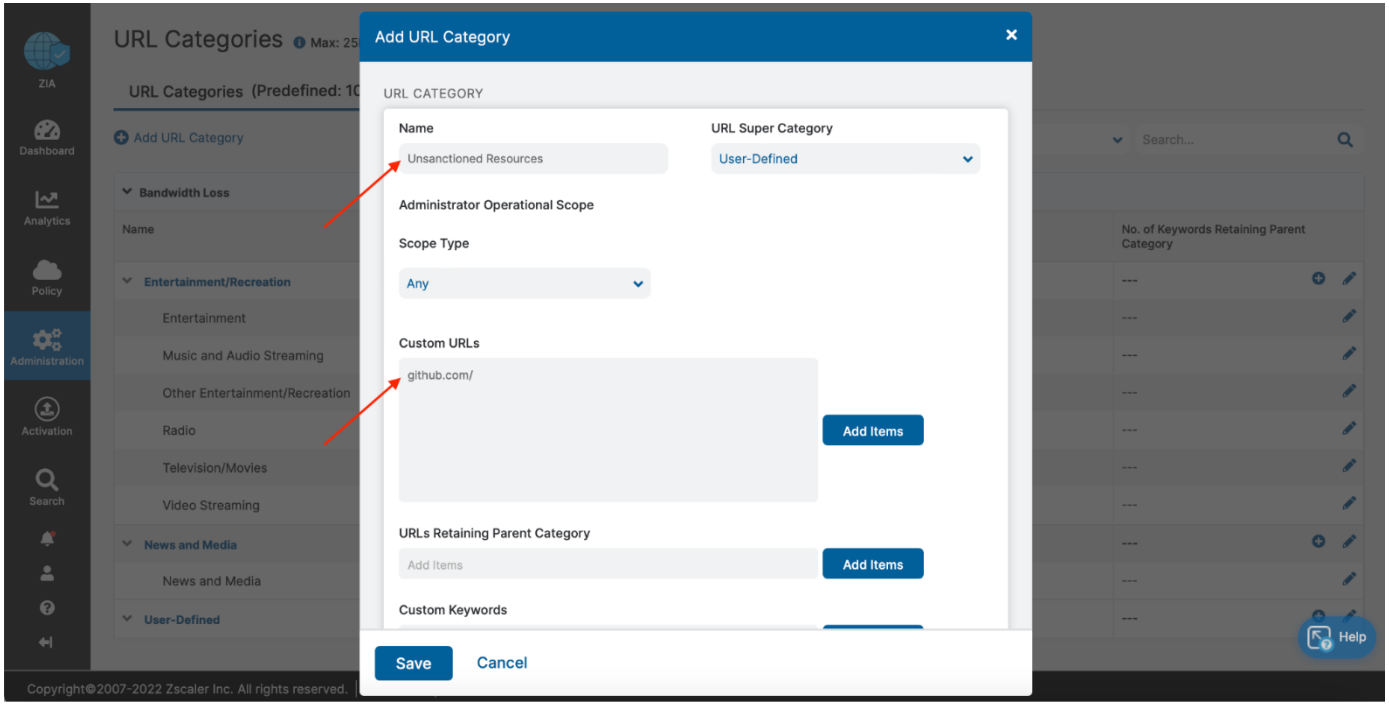


3. Configure the category as follows:
   a. For **Name**, enter **Sanctioned Resources**.
   b. Under **Custom URLs**, enter **github.com/aws-samples/** and **github.com/aws/**.



> ***Note:*** *Make sure to include the trailing "/".*

4. Click **Save**.
5. Click **Add URL Category** again.
6. Configure the category as follows:
   a. For **Name**, enter **Unsanctioned Resources**.
   b. Under **Custom URLs**, enter **github.com/**.

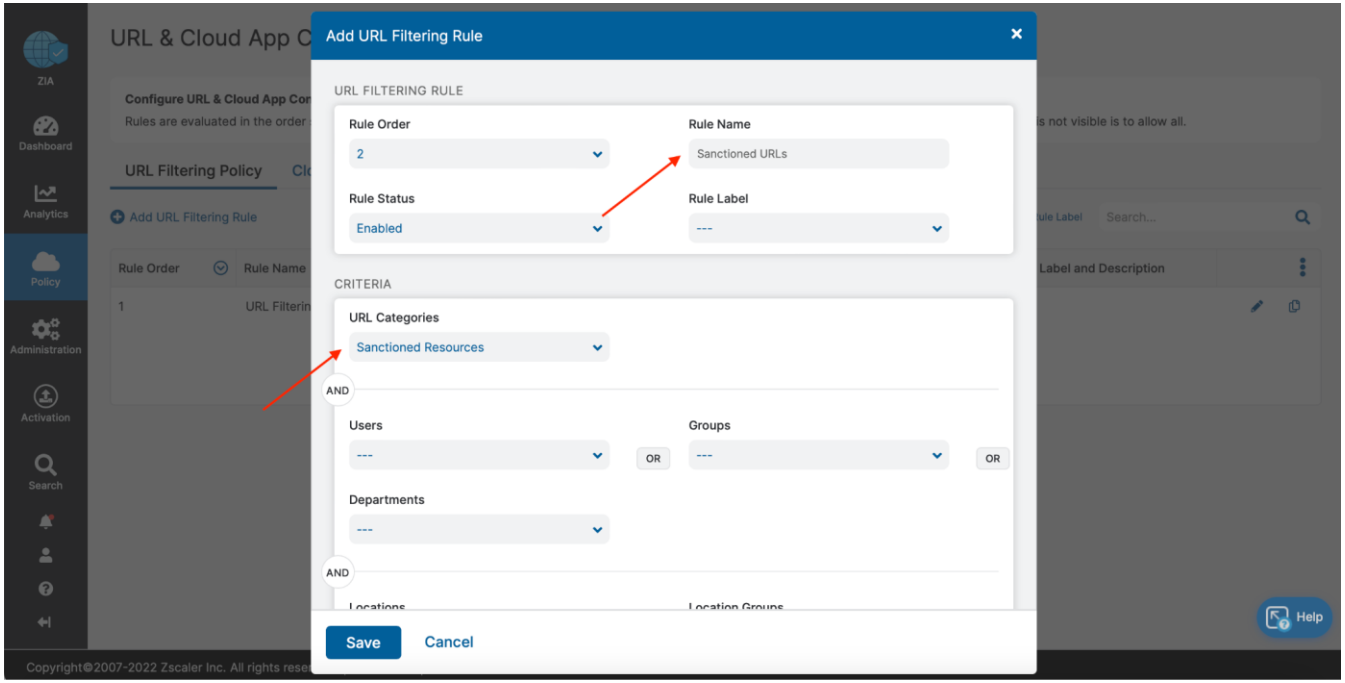


> ***Note:*** *Make sure to include the trailing "/".*

7. Click **Add Items**.
8. Click **Save**.
9. Activate the changes.

Note: *After defining the two URL categories, it's necessary to add URL Filtering rules that leverage the two categories.*

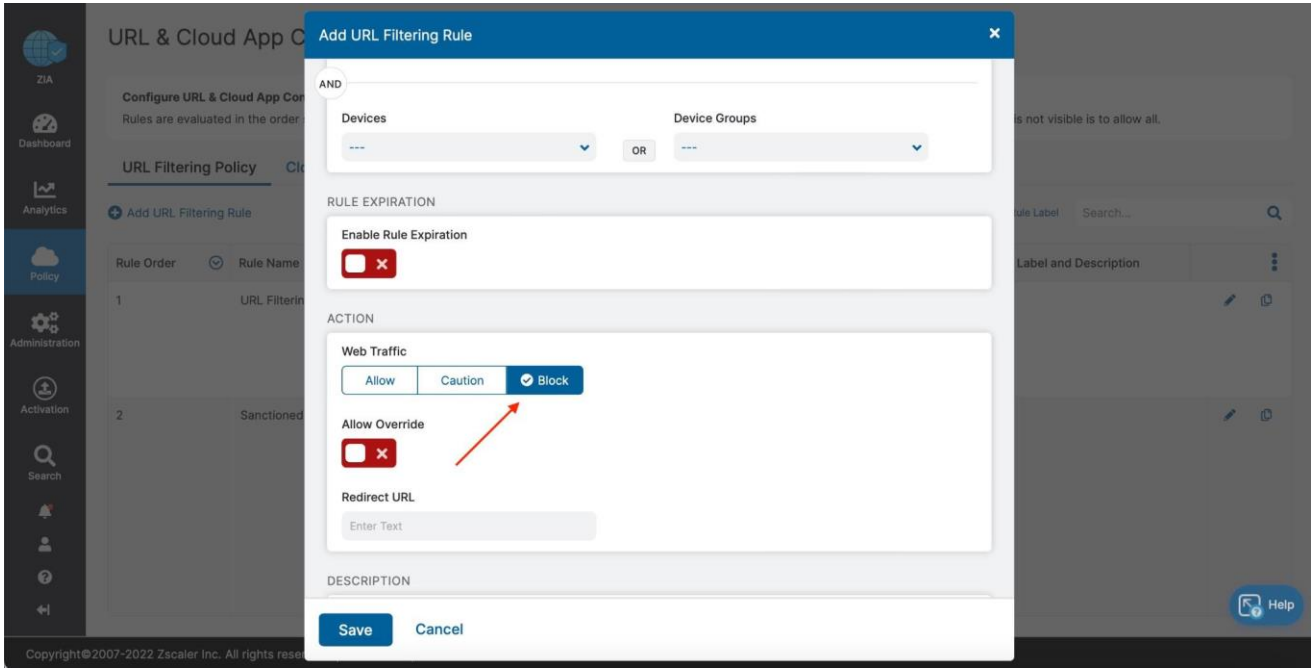
10. Select **Policy > ACCESS CONTROL > URL & Cloud App Control**.
11. To create a rule that allows access to sanctioned resources, click **Add URL Filtering Rule**.
12. Configure the rule as follows:
    a. For **Name**, enter **Sanctioned Resources**.
    b. In the Criteria section, under **URL Categories**, select **Sanctioned Resources**.
    c. From the **Locations** dropdown, select your **Region1** and **Region2** Locations.



13. Click **Save**.
14. Activate the changes.

> ***Note:*** *Similarly, add a new URL Filtering rule for unsanctioned resources that uses the Unsanctioned Resources URL Category and the same Location.*

15. Click **Add URL Filtering Rule** again.
16. Configure the rule as follows:
    a. For **Name**, enter **Unsanctioned Resources**.
    b. In the Criteria section, under **URL Categories**, select **Unsanctioned Resources**.
    c. From the **Locations** dropdown, select your **USEast2** Location.
    d. In the Action section, select **Block**.



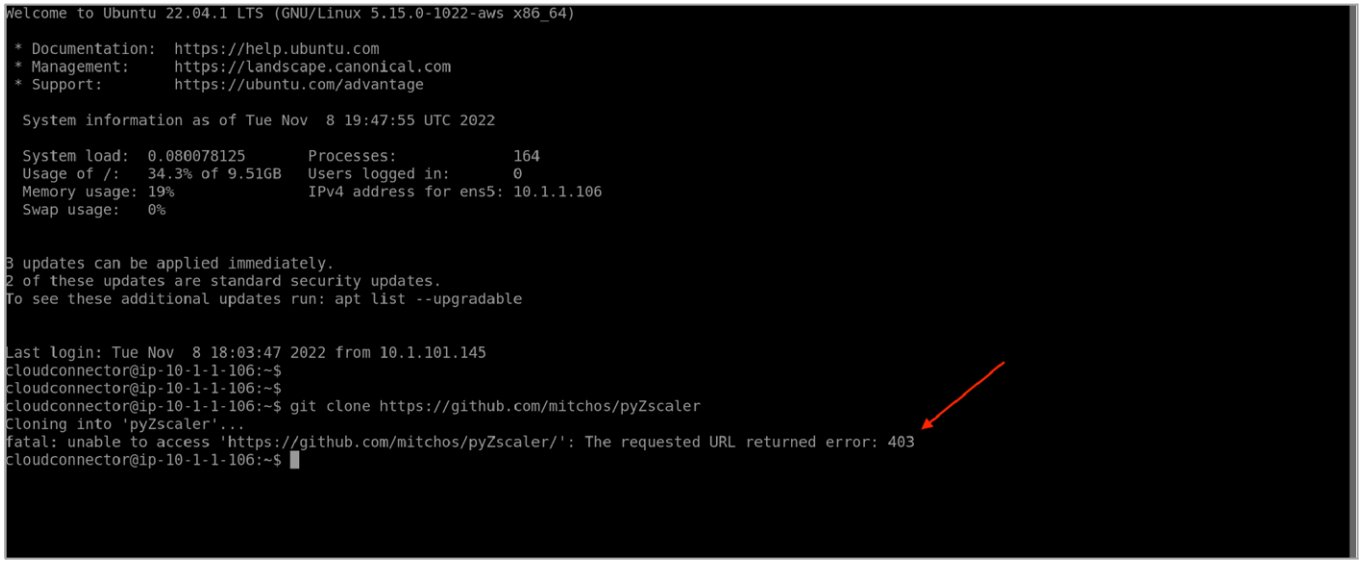
17. Click **Save**.
18. Activate the changes.

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

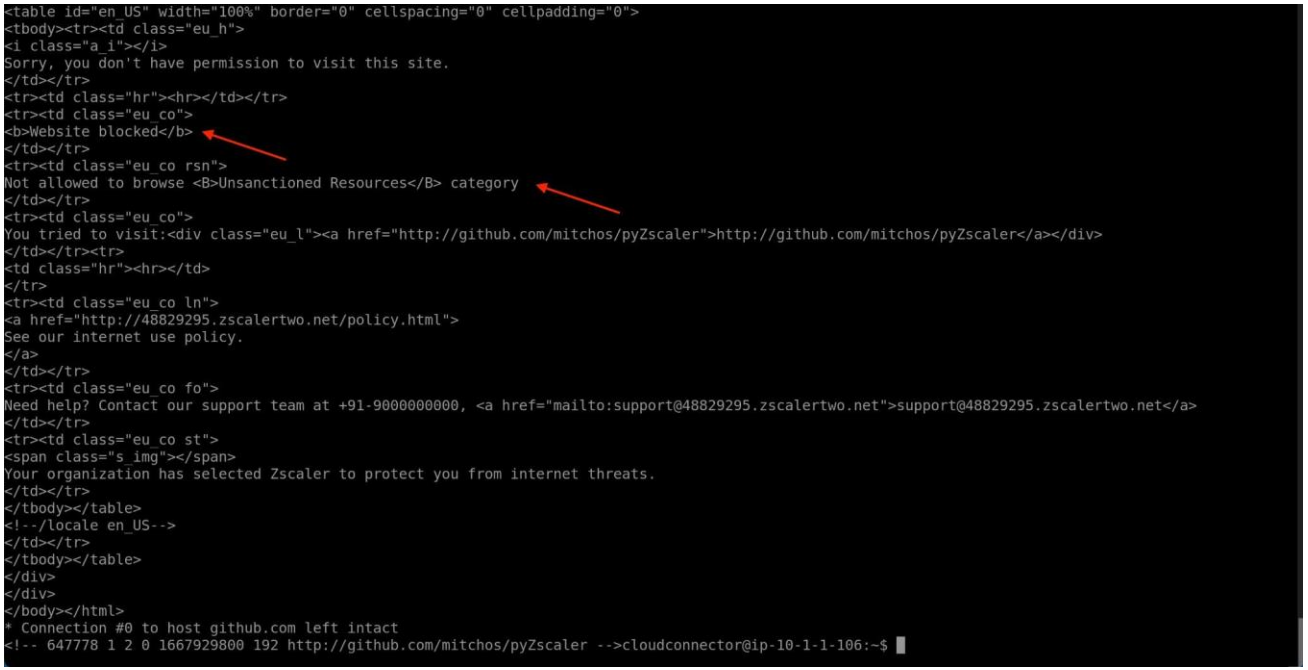
Task 2: Verify URL Filtering Rule

In this task, you will verify that the configured URL Filtering Rules have the desired effect.

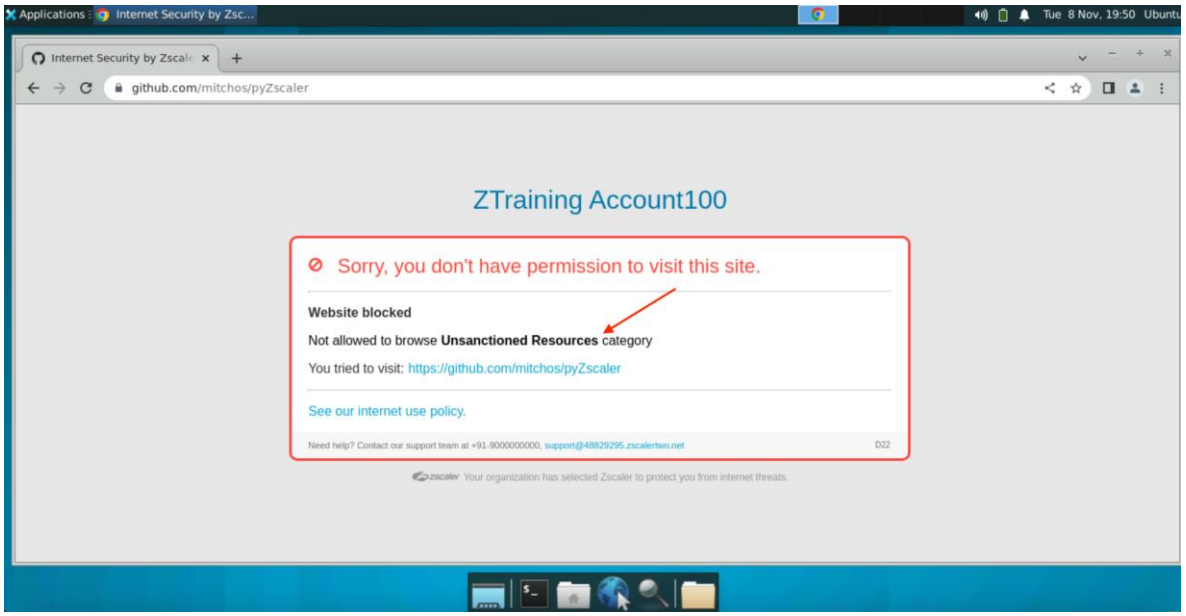
1. Return to the **Region1** Workload connection and open a terminal.
2. At the Command Line, enter **git clone https://github.com/mitchos/pyZscaler**.
3. Verify that Git returns an error (403, Unauthorized).



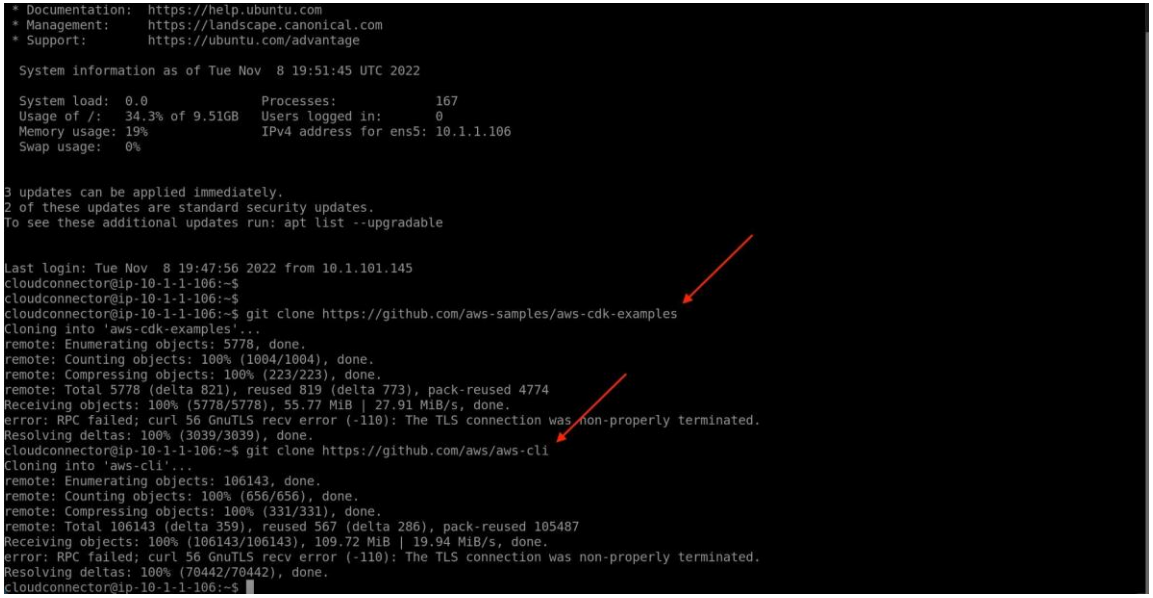
Note: *You can also use* ***curl -v github.com/mitchos/pyZscaler*** *to simulate a web browser when accessing this URL for additional information.*



4. Alternatively, you can also go to this website via Chrome to verify the behavior as well.



5. To clone the aws-cdk-examples and aws-cli repositories, enter **git clone https://github.com/aws-samples/aws-cdk-examples** and **git clone https://github.com/aws/aws-cli**.
6. Verify that the cloning of any repositories under aws-samples and aws was successful.



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

Lab 9: Integrating with Zscaler Private Access

During mergers and acquisitions, the acquiring company and the acquired company often need access to each other's applications - such as databases, web servers or financial tools. The traditional approach to achieve this goal is to connect the networks together. But this approach has two major drawbacks:

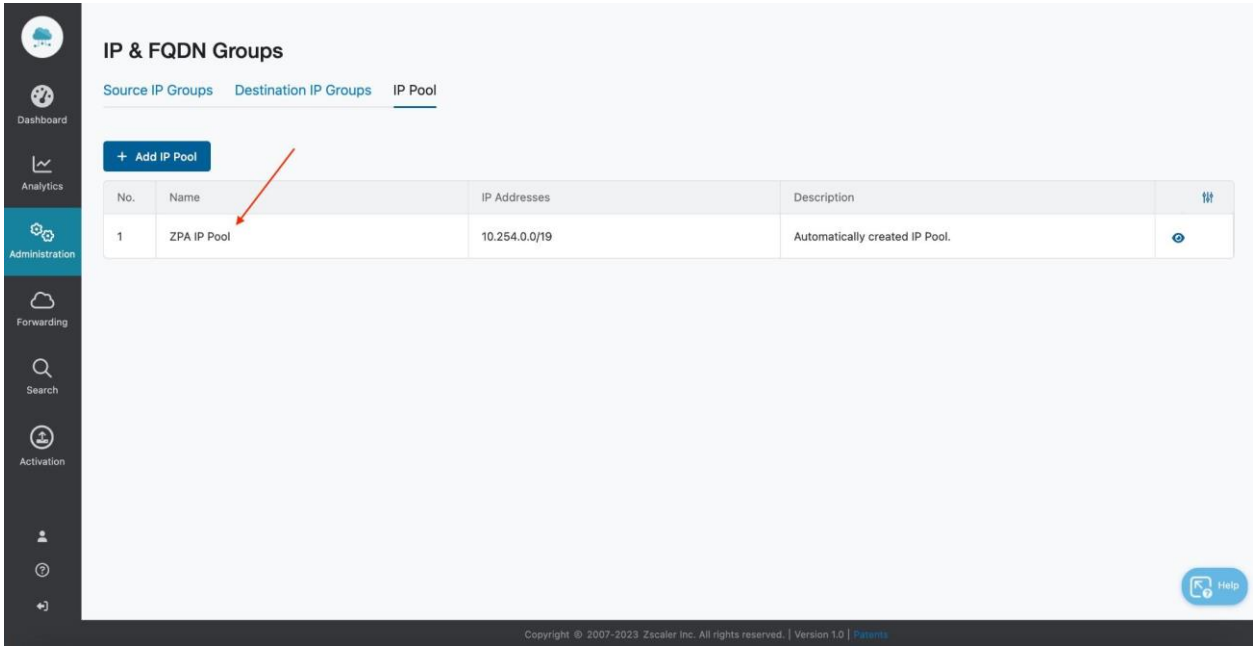
1. There is a high likelihood that the IP Address space may overlap. In which case, additional technologies, such as Network Address Translation, must be used to overcome this.
2. The security standard between the two companies differs, meaning they do not trust each other and therefore must go through an extensive vetting process.

It is not uncommon for this IT integration process to take 12 months or more to complete. This often introduces friction between IT and the business because, during this time, the business is not able to access key infrastructure of the acquired company. This module demonstrates how easy it is to overcome the two challenges mentioned above using Cloud Connector and Zscaler Private Access.

Note: *Cloud Connectors integrate with ZPA to allow workloads to seamlessly connect to other resources within your organization, or an acquired organization. To proxy this traffic, workload DNS queries are intercepted by the Cloud Connector. The Cloud Connector then responds to the DNS request on behalf of the "real" DNS server using a Synthetic IP. This forces the cloud workload to send traffic for the intended destination towards the Synthetic IP hosted on the Cloud Connector. On the receiving end of this transmission, the destination workload receives the traffic from the remote App Connector IP Address, rather than the "real" IP Address of the source workload.*

---

1. From the Cloud Connector portal, navigate to the **Administration** menu, followed by **IP & FQDN Groups**.
2. Click the **IP Pools** tab.

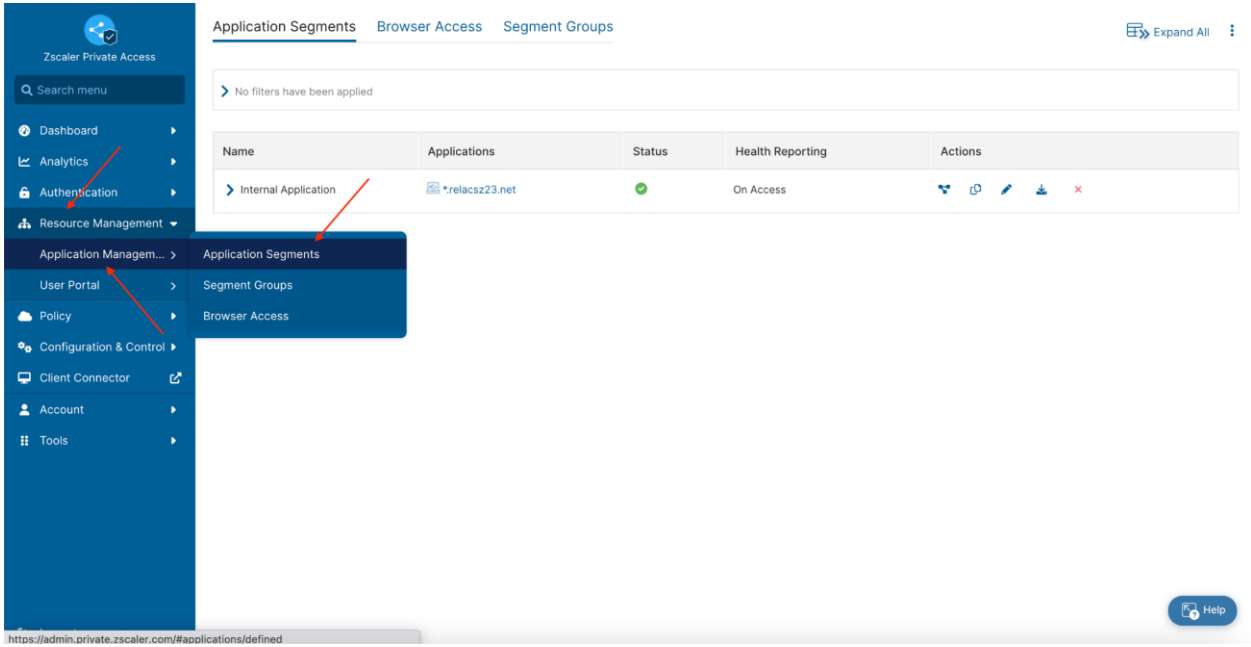


Here, we can see the Synthetic IP Pool that Cloud Connector appliances will use to proxy ZPA-bound traffic. Let's create some ZPA Application Segments so we can see this in action.

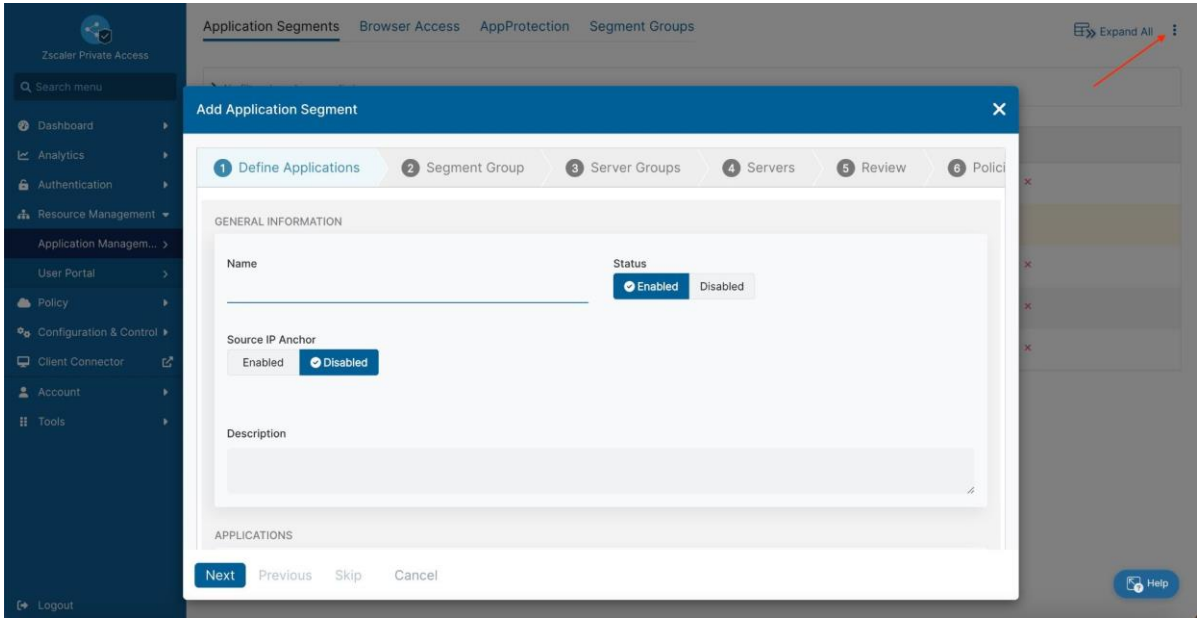
Note: *The focus of this hands-on lab is on Cloud Connector. Hence, App Connectors and their deployment will not be covered and are already operational in your lab. Likewise, an in-depth discussion around Access Policy, Client Forwarding Policy or ZPA architecture in general is not covered.*

---

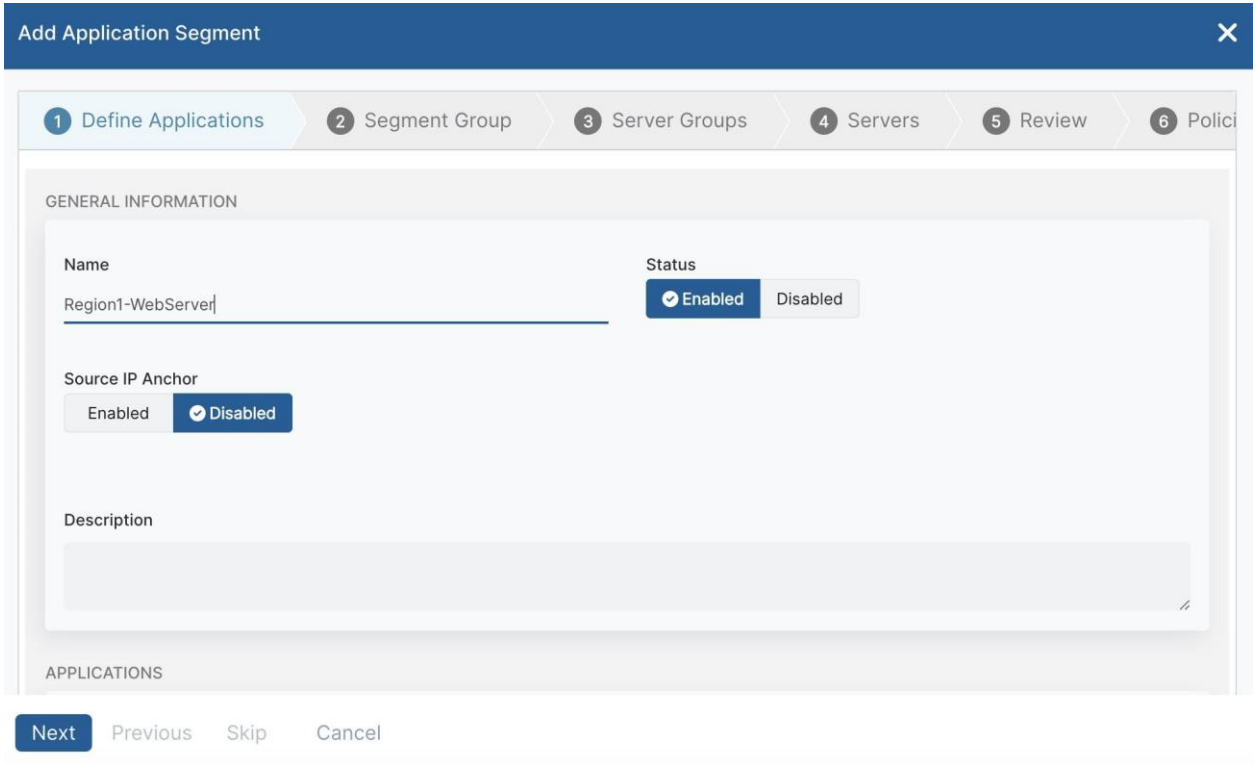
3. Navigate to the ZPA Admin portal and login with your pod's credentials.
4. In the Resource Management menu, hover over the **Application Management** menu item, then select **Application Segments**.



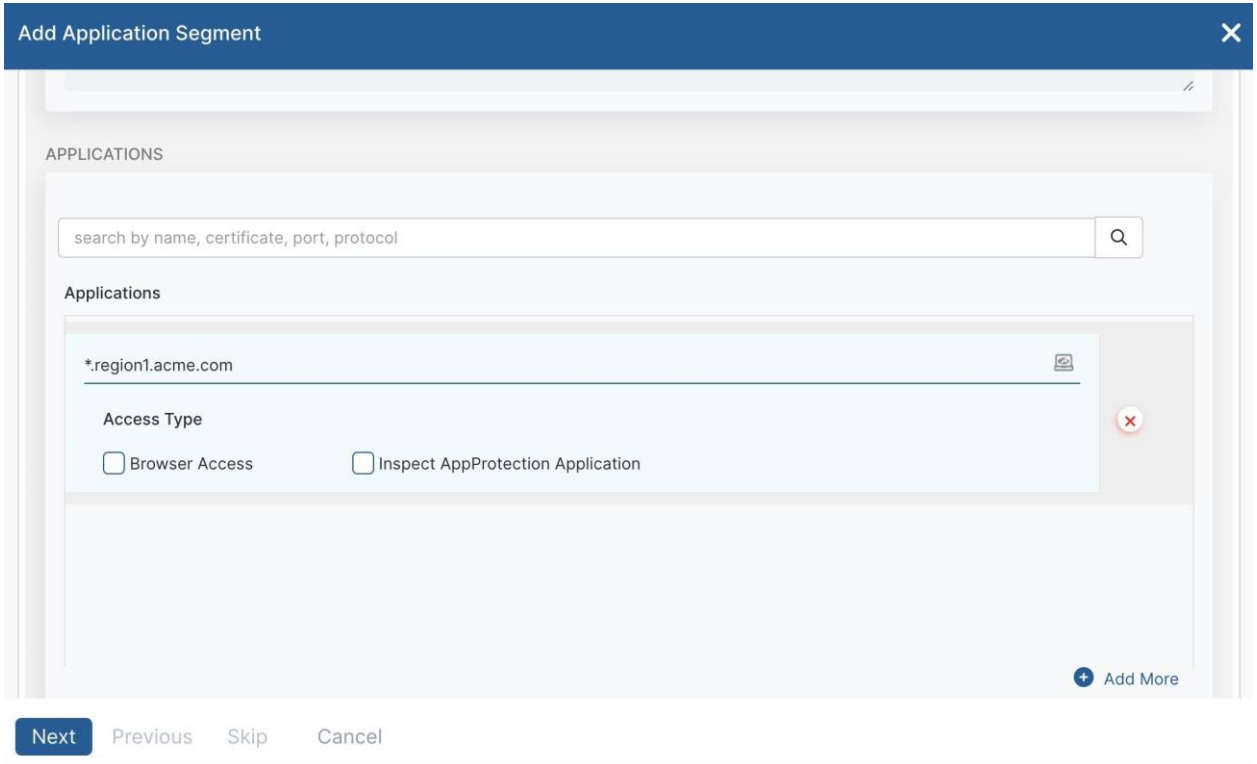
5. Click the **three dots** in the upper right corner of the screen and choose **Add Application Segment**.



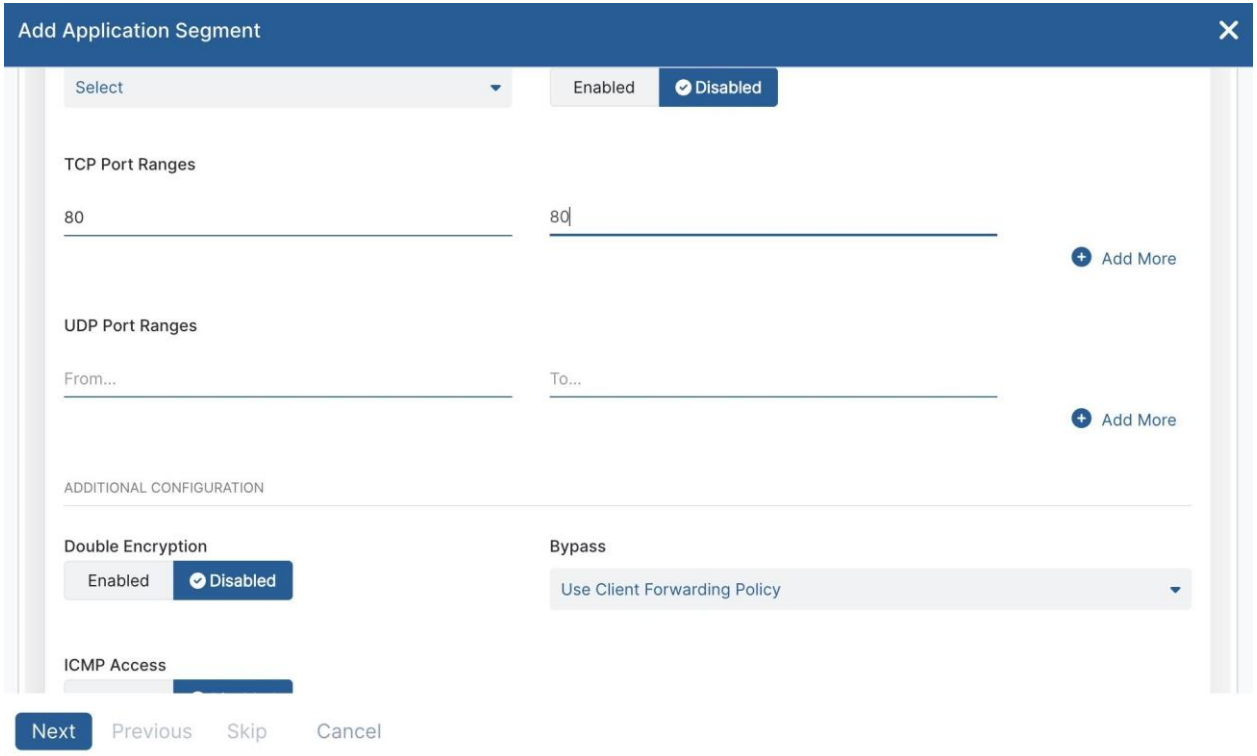
6. Provide a **name** for the Application Segment. In our case, we're going to test with a simple web-based file server, so we'll name our application **Region1-WebServer**.



7. Enter the **domain** that you wish to proxy under the Applications section. Here, we'll use ***.region1.acme.com**.



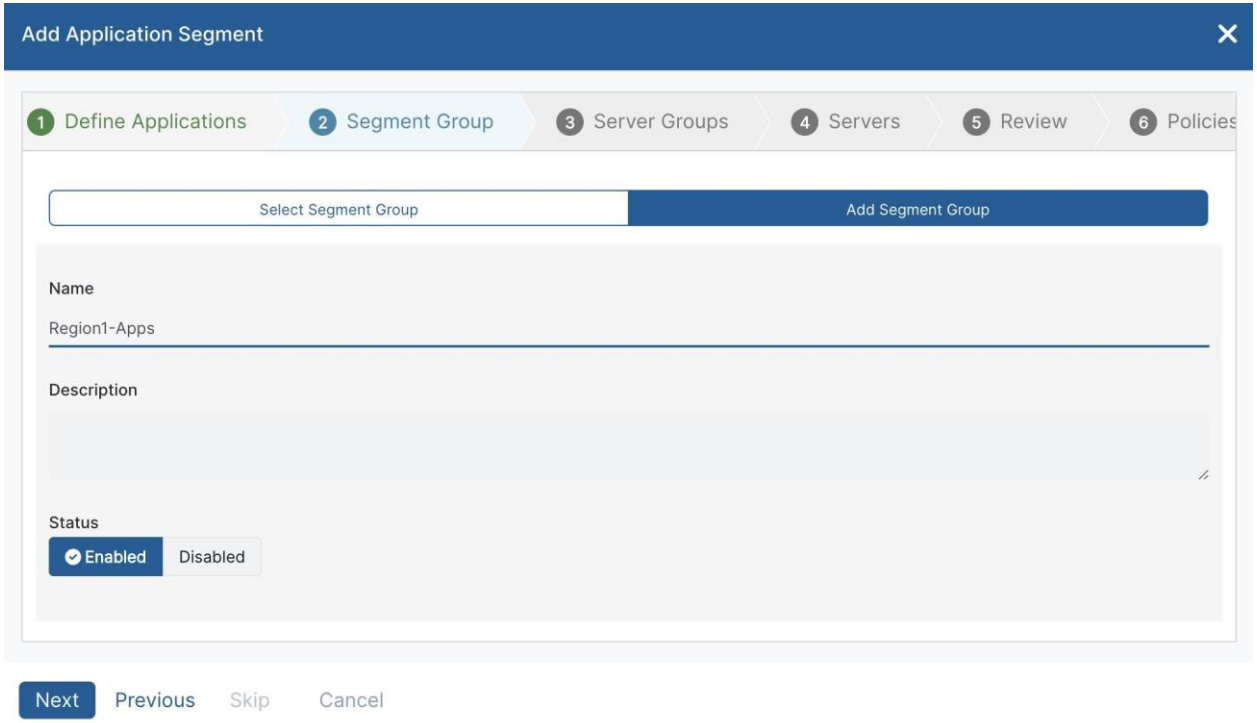
8. In the **TCP Port Ranges** section enter port **80**, as shown below, to define our web traffic.



9. Click the **Next** button to proceed to the next step.

Segment Groups allow administrators to bundle multiple Application Segments together for easier policy definition. Rather than permitting or denying access to individual Application Segments, a policy can permit or deny access to a Segment Group, which contains multiple Application Segments. In our case, we'll create a new Segment Group called Region1-Apps.

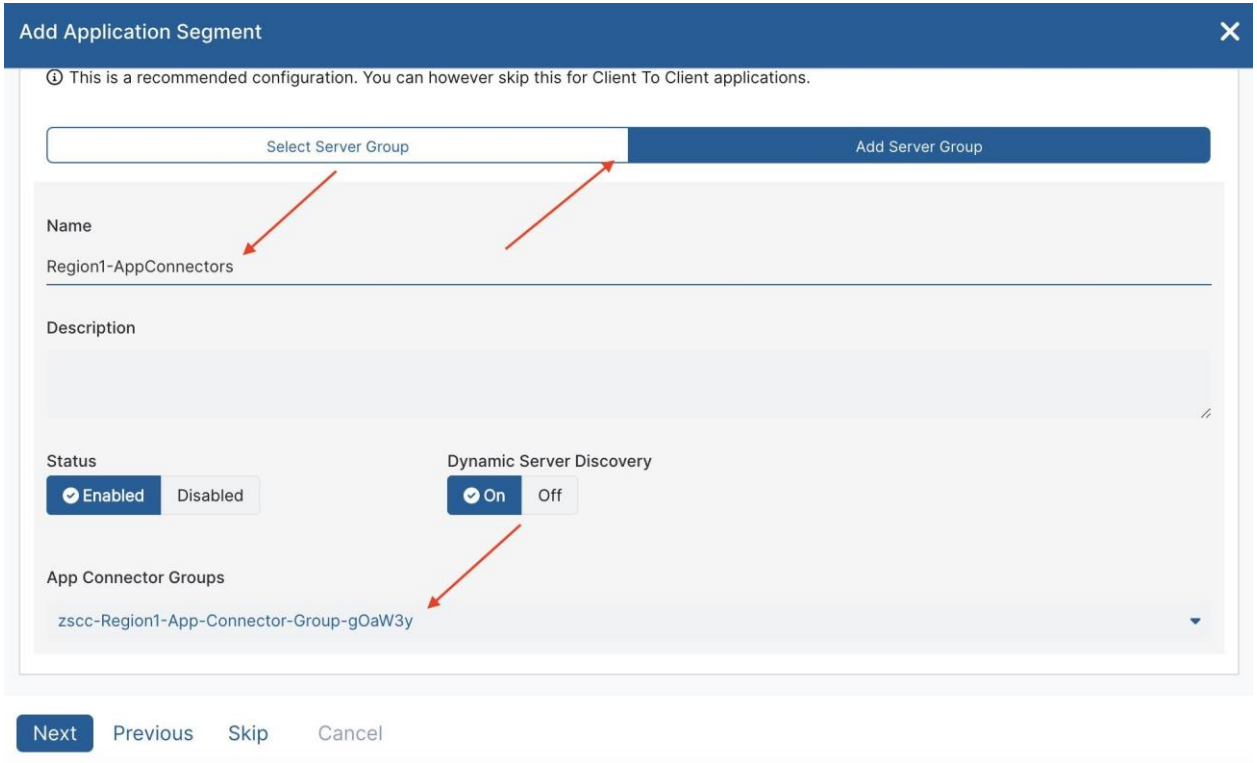
10. Click the **Add Segment Group** tab and input your Segment Group name **Region1-Apps**.



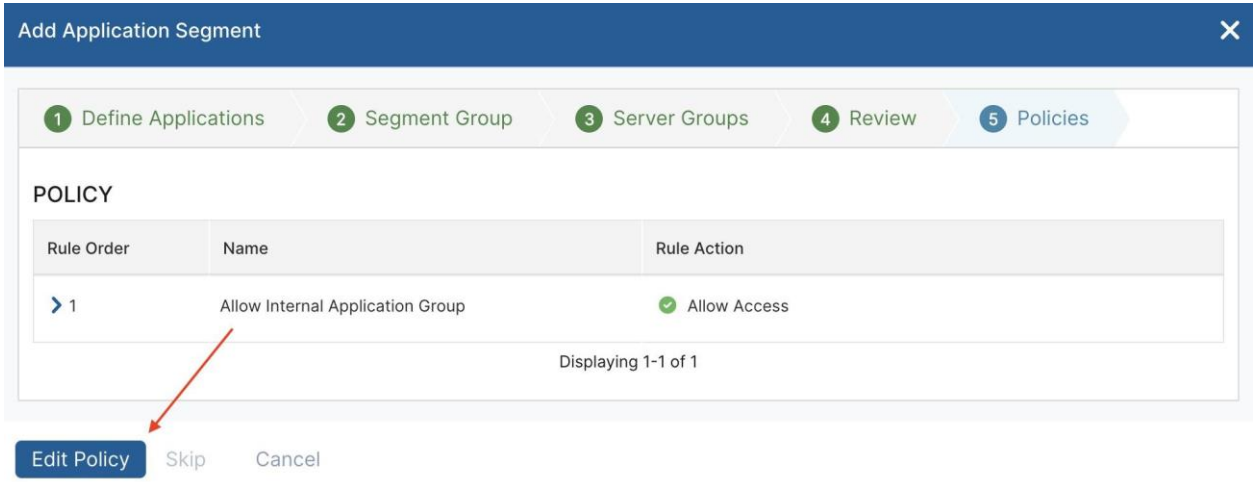
11. Click the **Next** button to proceed.

Server Groups allow an administrator to define which App Connectors, or groups of App Connectors can reach a specific application. Here, we'll create a new Server Group that specifies our deployed App Connector.

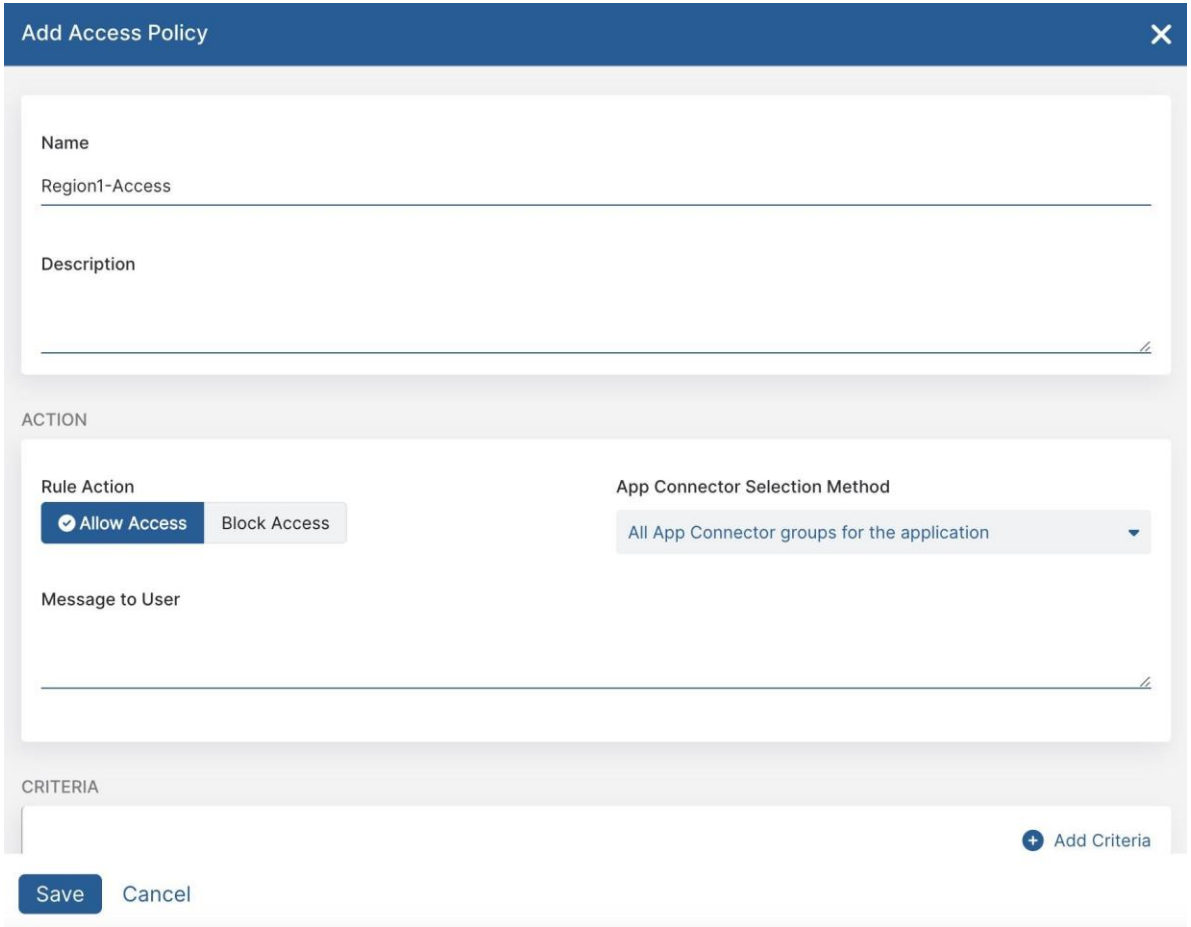
12. Click the **Add Server Group** button, enter the **name** of your Server Group, then select your **App Connector** in the dropdown menu at the bottom.



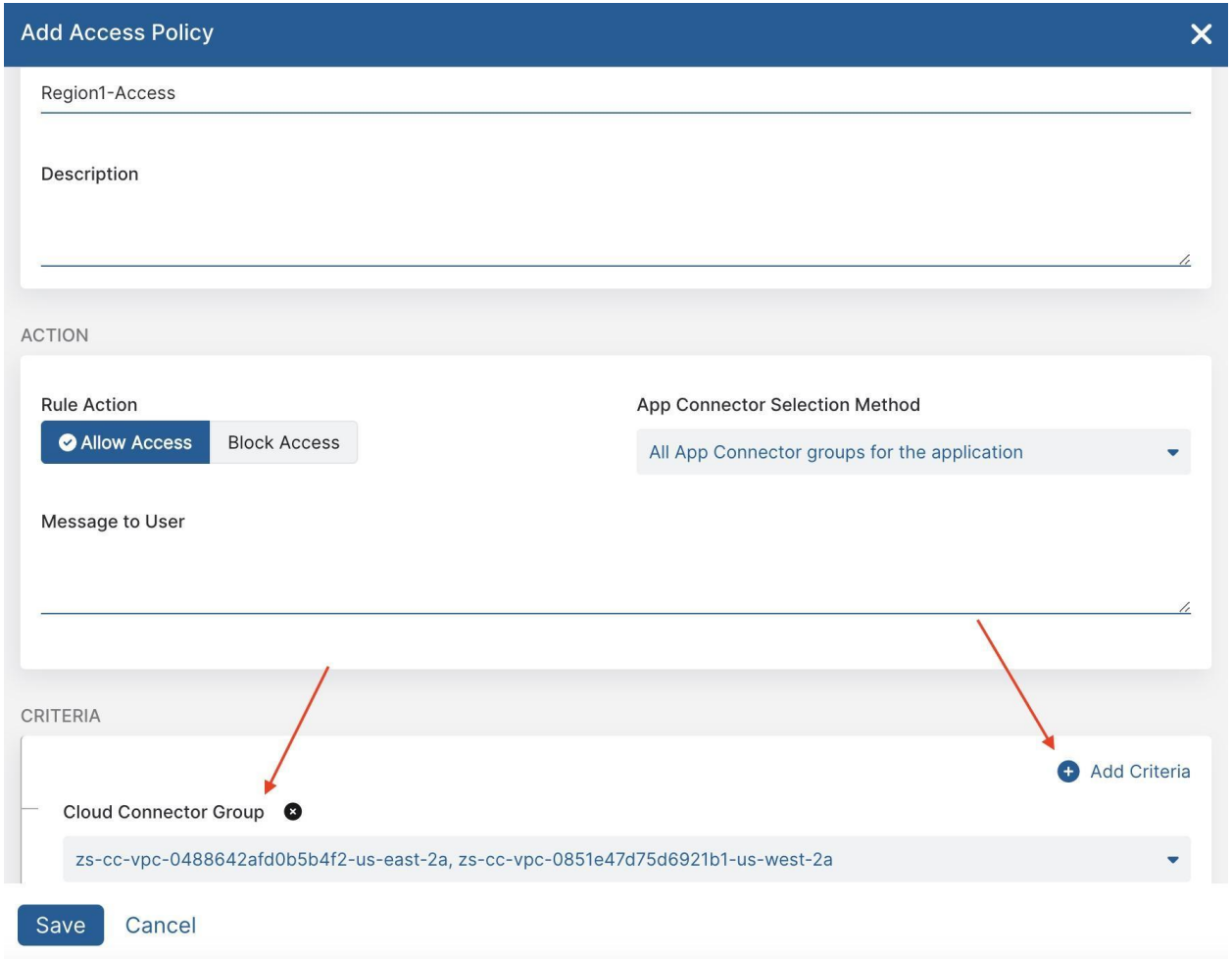
13. Click the **Next** button to proceed, then **Save** your new Application Segment. The workflow should then prompt you to edit the Access Policy to allow traffic.
14. Click the **Edit Policy** button.



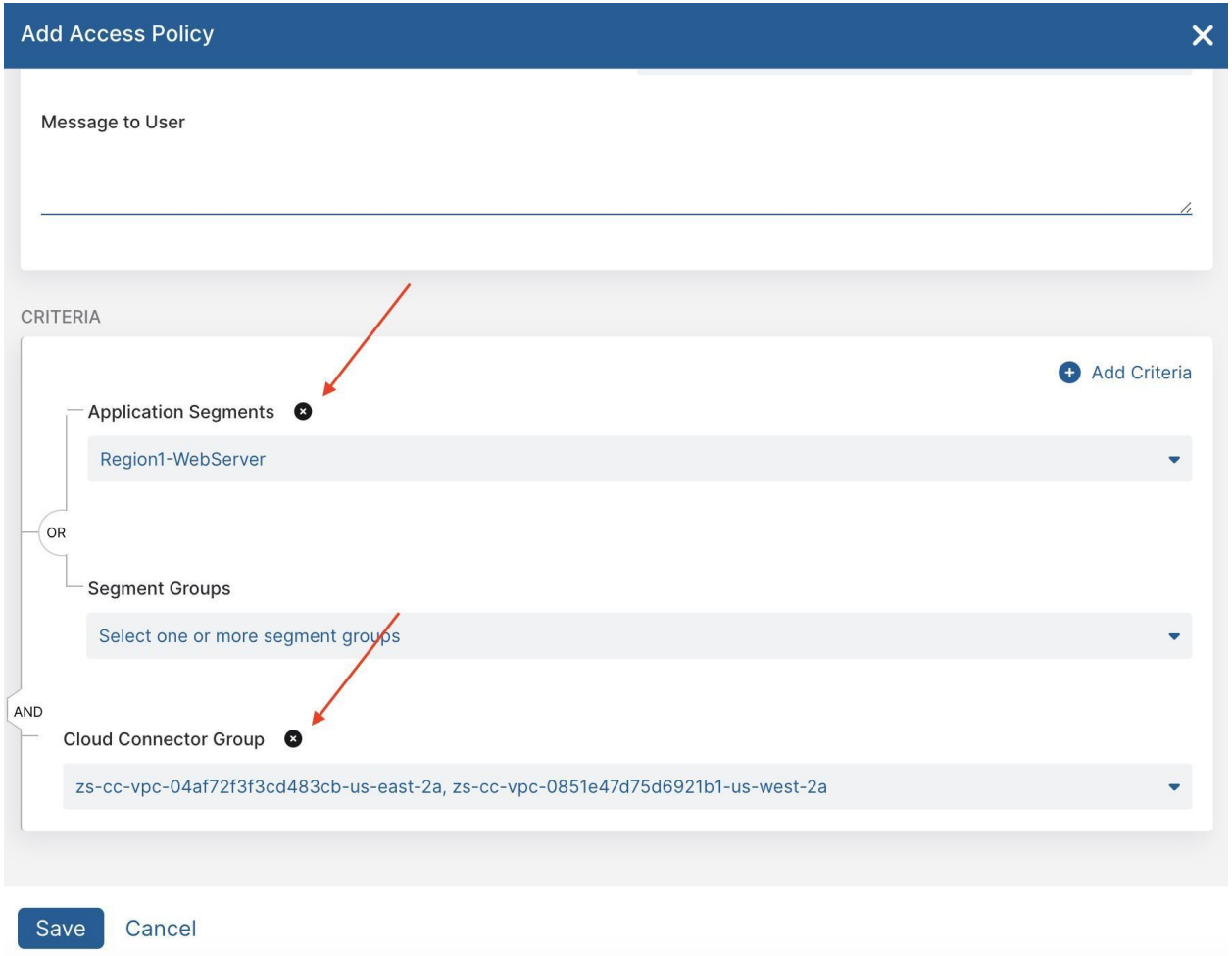
15.Click the **Add Rule** link in the top right corner of the page.
16.In the **Add Access Policy** workflow, give your Access Policy a **name** (such as Region1-Access).



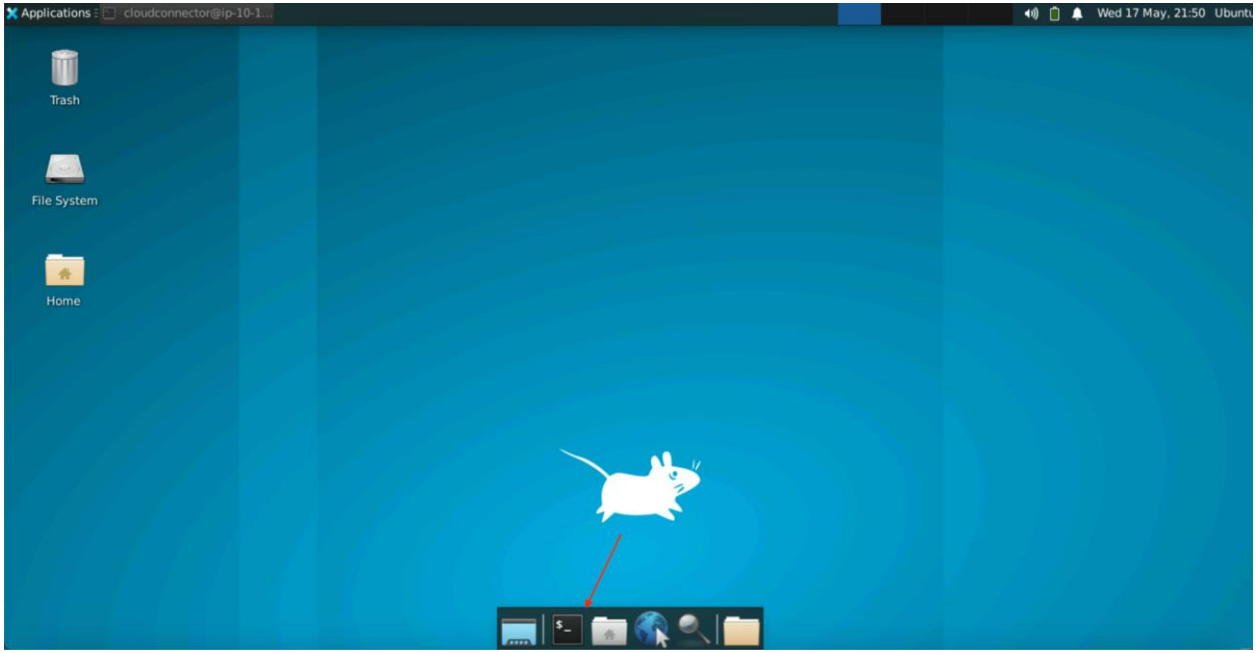
17. In the **Criteria** section, click the button to **Add Criteria**, then choose **Cloud Connector Groups**.
18. In the dropdown that appears, choose your **Cloud Connector** appliances.



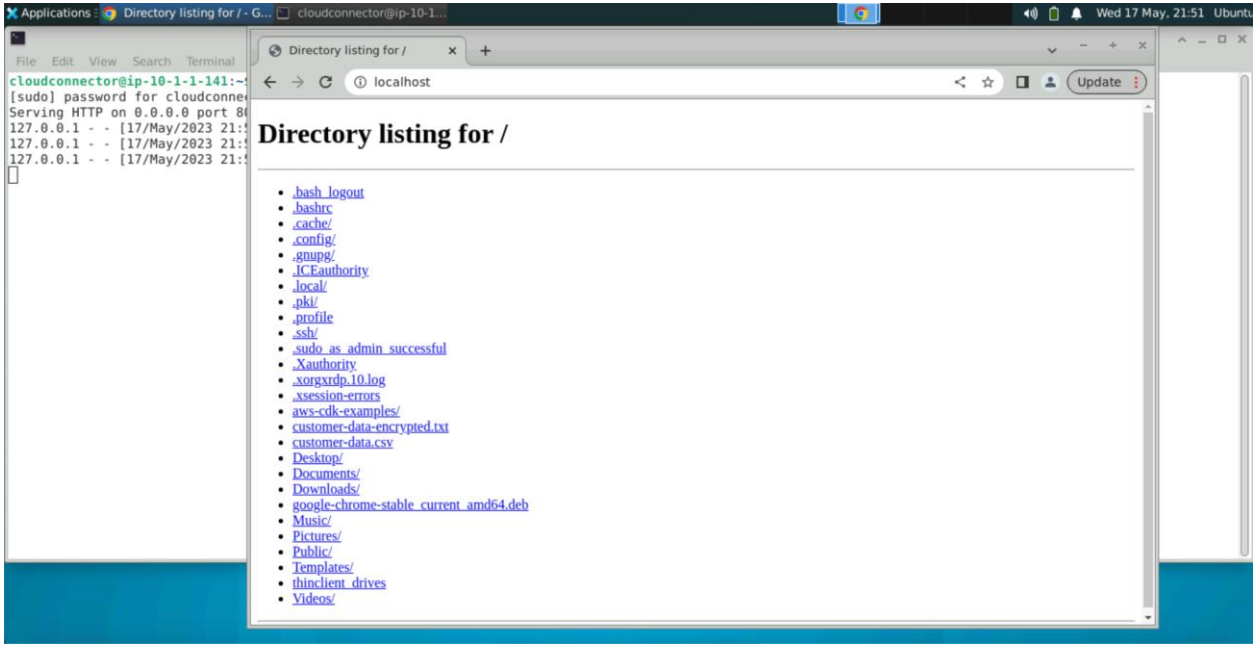
19.Click the **Add Criteria** button again, this time adding an Application. Choose your **Application Segment**.



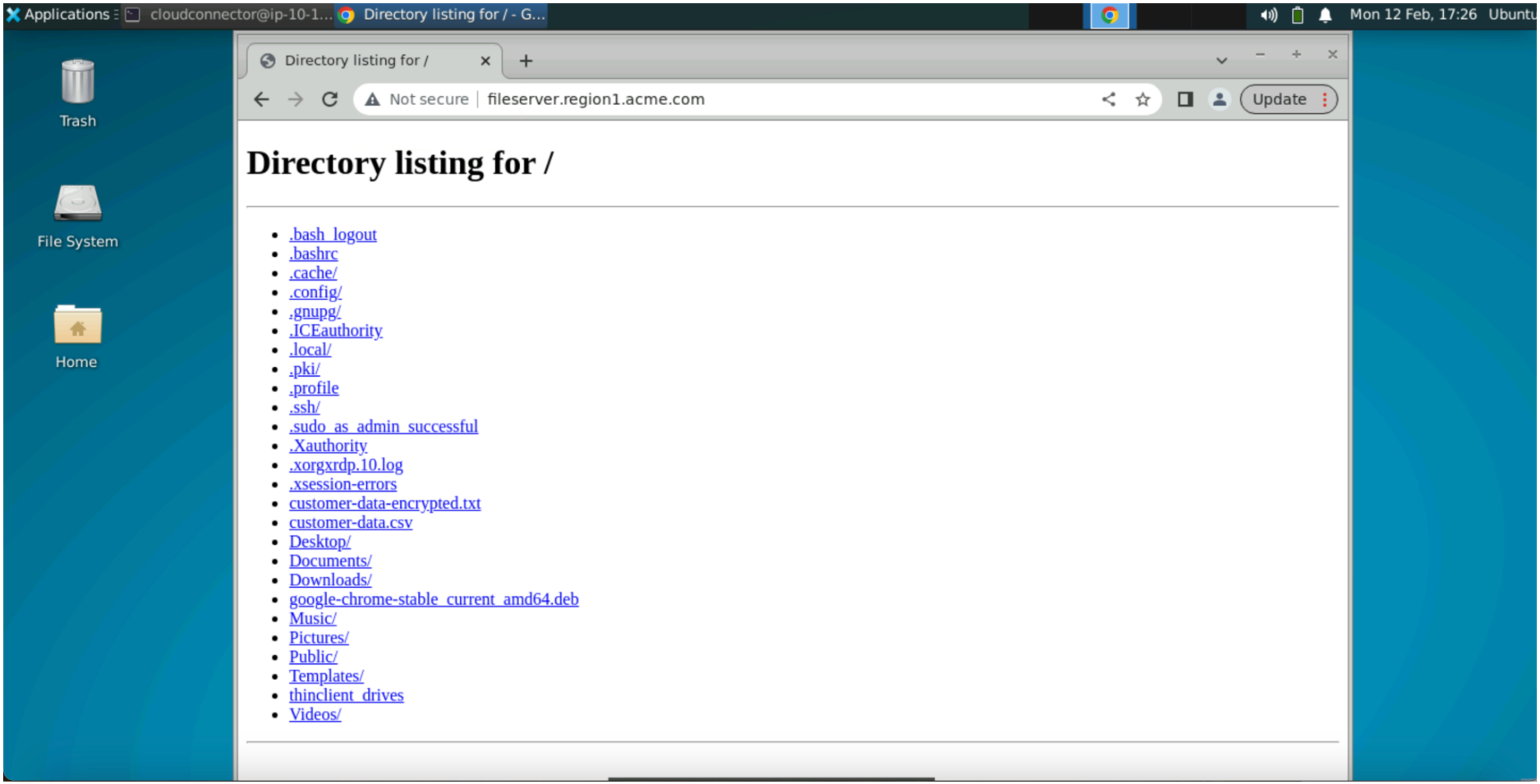
20.Next, it's time to test. Navigate to your **Region1** workload and open a Terminal.



21. We'll use a simple Python-based web server for this test. Enter the command **sudo python3 -m http.server 80**. You may be prompted for a password. If so, enter your workload's sudo password: **CloudConnector2022!**.
22. Once started, the server should provide access to the user's Home directory. You can test this by opening a web browser and browsing to **http://localhost**. This should display the contents of the Home directory.

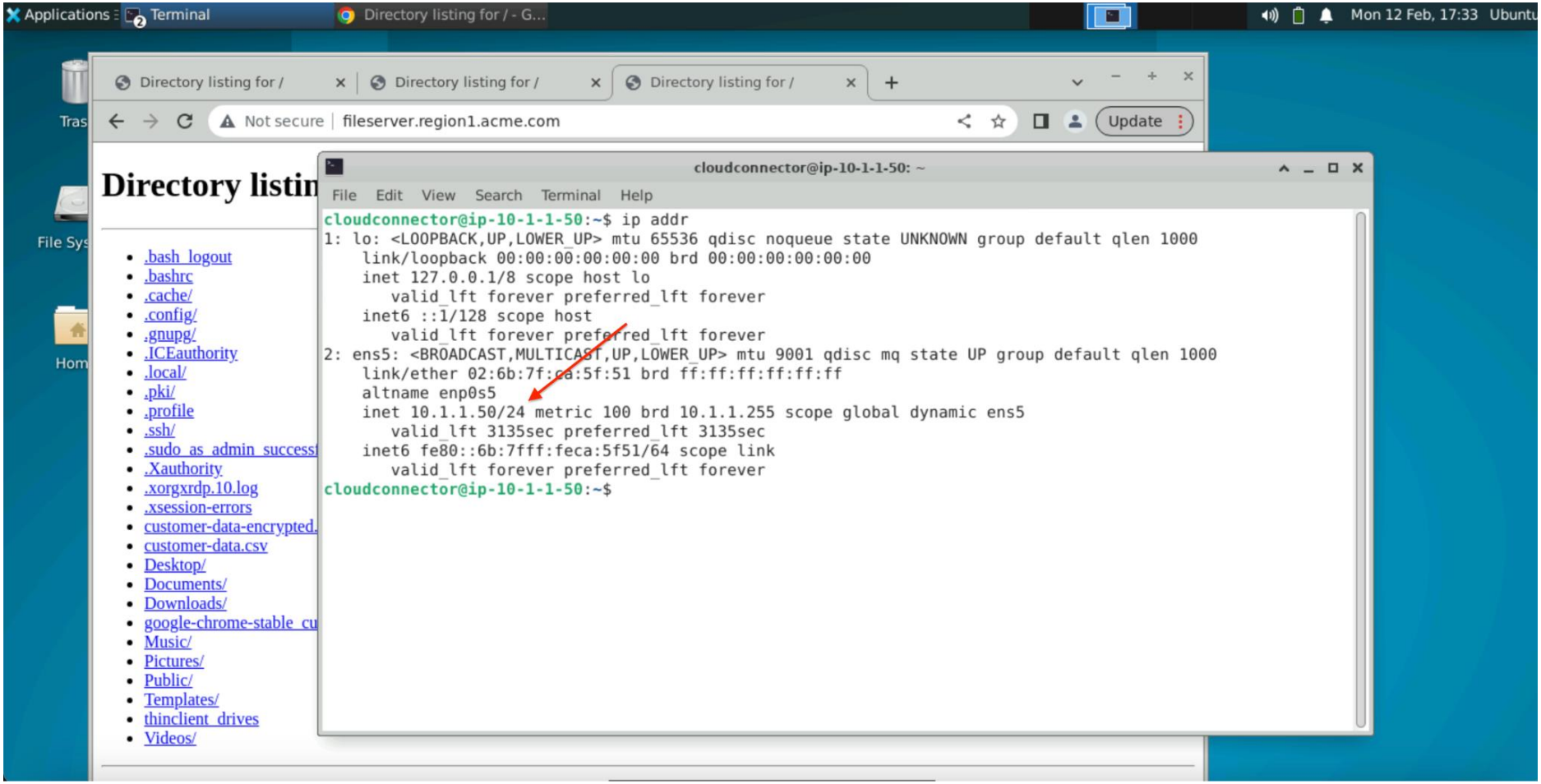


23.Navigate to your **Region2** workload.
24.Open a web browser and enter http://fileserver.region1.acme.com . This should display the Region1 workload's Home directory.

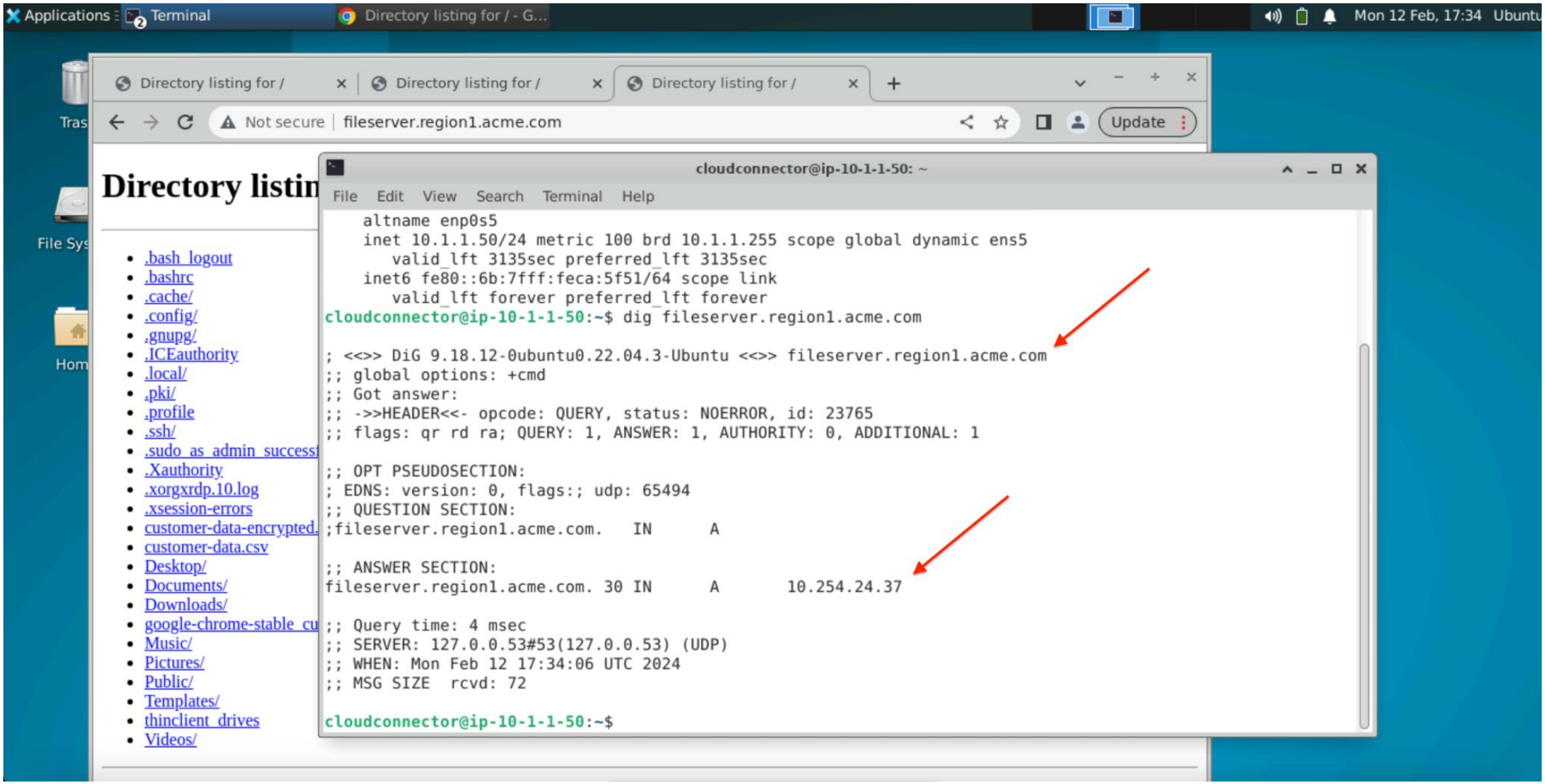


Congratulations! You've connected two workloads together over Zscaler Private Access!

25. What's interesting here, however, is how this traffic was proxied. Open a terminal on your **Region2** host, then enter the command **ip addr**. Note the IP Address of the Region2 machine.

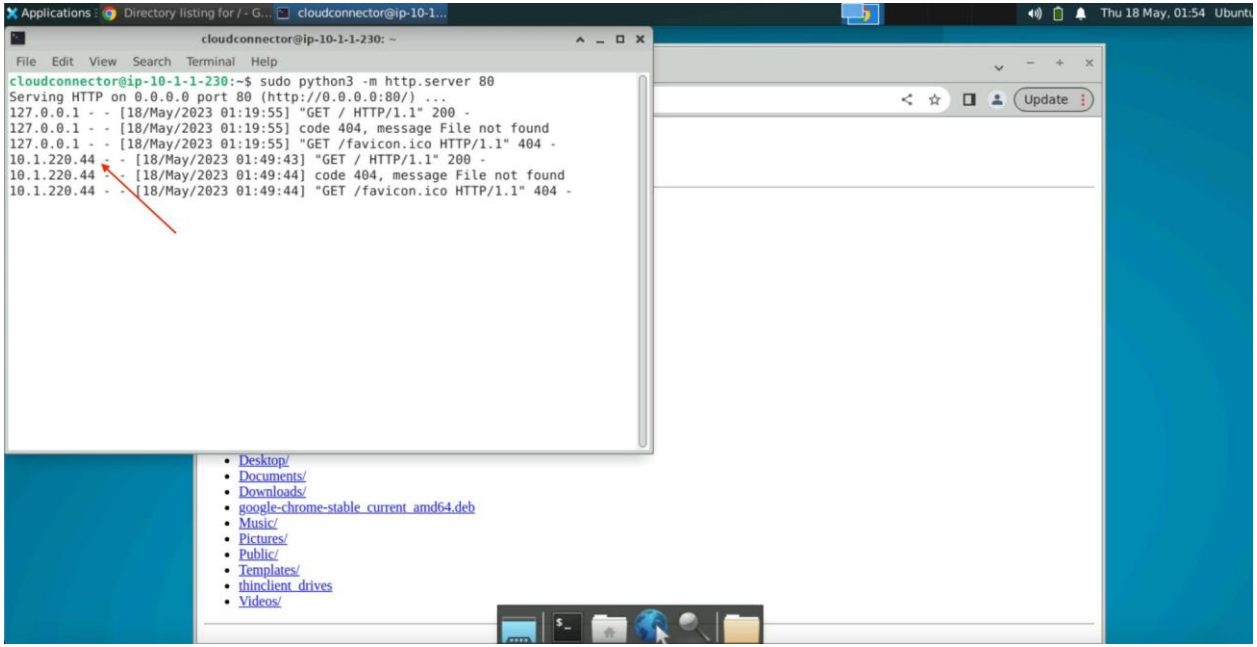


26. Similarly, enter the command **dig fileserver .region1.acme.com** from the terminal. Notice the IP Address that DNS returned for the remote web server.



Note: *The IP Address returned should be within the Synthetic IP Pool we viewed in the Cloud Connector portal at the beginning of this module.*

27. Now, navigate to your **Region1** workload. Review the activity in the web server terminal that was logged when the Region2 host accessed this machine over ZPA. Notice the IP Address the web server saw is that of the App Connector - not the “real” IP Address of the Region2 workload!



28. Furthermore, open a new Terminal on the **Region1** workload.
29. Enter the command **ip addr**.

Note: *If you recall, the IP Address that was returned in the dig command on the Region2 host was a different address. This means that the Region2 host is communicating with an IP Address that is not the “real” address of the destination web server! And likewise, the Region1 host was responding to an IP Address that was not the “real” source of the traffic. And, further, both workloads exist on the same subnet in two different Regions, meaning that ZPA handled the IP Address overlap for us!*

As you can see, this entire communication was brokered through Synthetic IP Addressing via Cloud Connector and App Connector. These two regions were never directly connected, nor did “real” IP Address ranges ever exchange. This can make multi-cloud connectivity a breeze - especially when overlapping IP ranges are in place, such as with Mergers and Acquisitions or Divestitures!

Summary

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.

Clean-up

Thank you for participating in the Zscaler for Workloads Hands-on Workshop! To help make the hands-on pods available for future users, please complete the following steps prior to logging out completely from your pod. This will ensure clean-up scripts can fully reset the environment prior to the next session:

1. Remove your ZPA Access Policy from the **Policy > Access Policy** page.
2. Remove your ZPA Segment Group from the **Resource Management > Application Management > Segment Group** page.
3. Remove your ZPA Application Segment from the **Resource Management > Application Management > Application Segments** page.
4. Remove your ZPA Server Group from the **Configuration & Control > Private Infrastructure > Server Groups** page.