App Delivery Manager

Lab Guide

TechXchange 2023

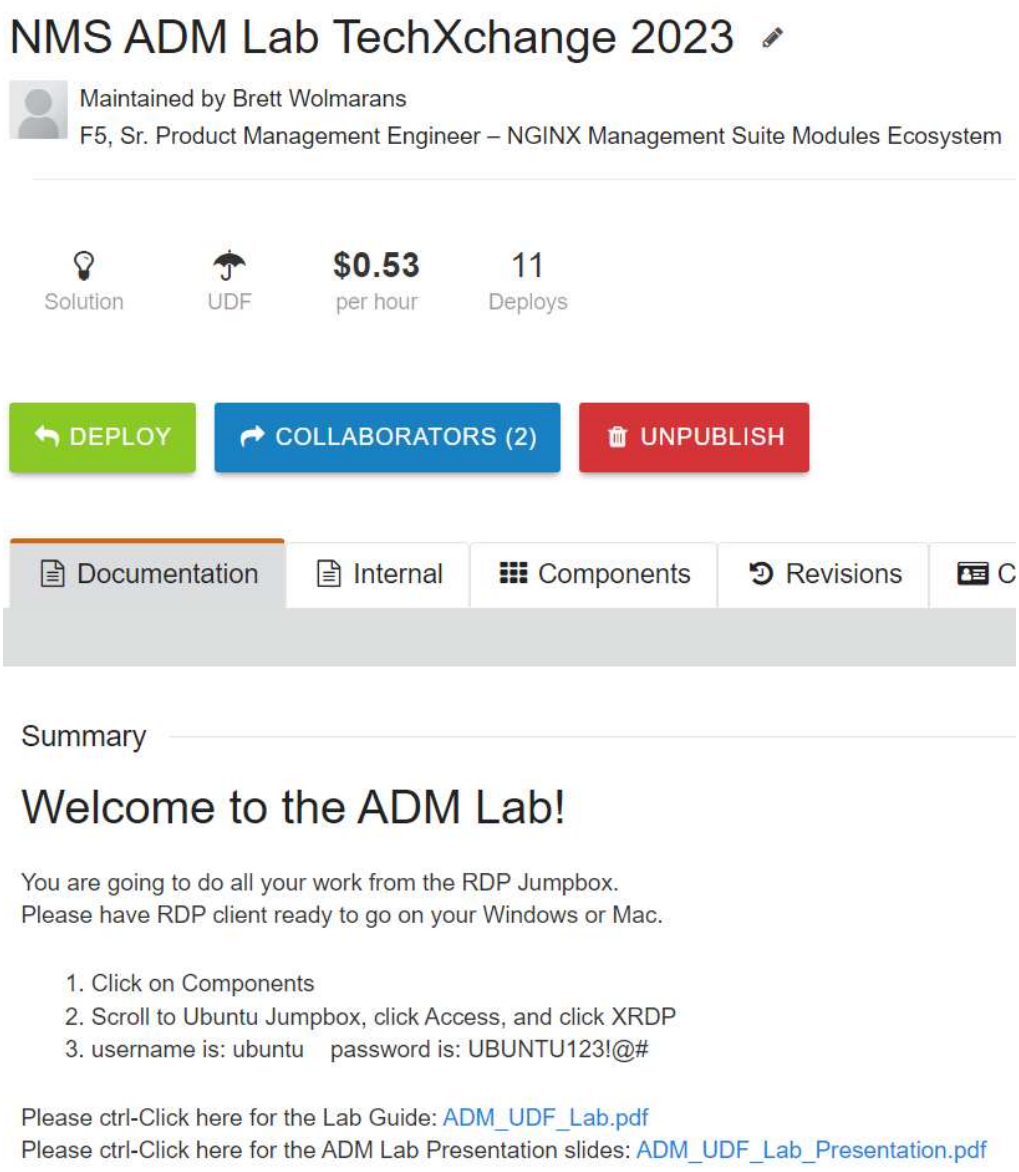
Version 1.2

6/28/2023 3:57 PM

UDF Lab

Deploy the following UDF Blueprint **in the region closest to you**:

1. Click on Components
2. Scroll to Ubuntu Jumpbox, click Access, and click XRDP
3. username is: ubuntu password is: UBUNTU123!@#



Presentation Slides

The Lab Presentation Slides link can be found in the UDF deployment documentation.

Module 1 – Simple HTTP Deployment

Learn how to create a simple HTTP deployment that securely load balances between several apps including microservices apps

High Level Business Objective

- You work in the information technology department for a global drinks company, and you have been tasked with delivering the following critical business applications reliably, securely, and with high performance:
  - **Brewz**
  - **Juiceshop**

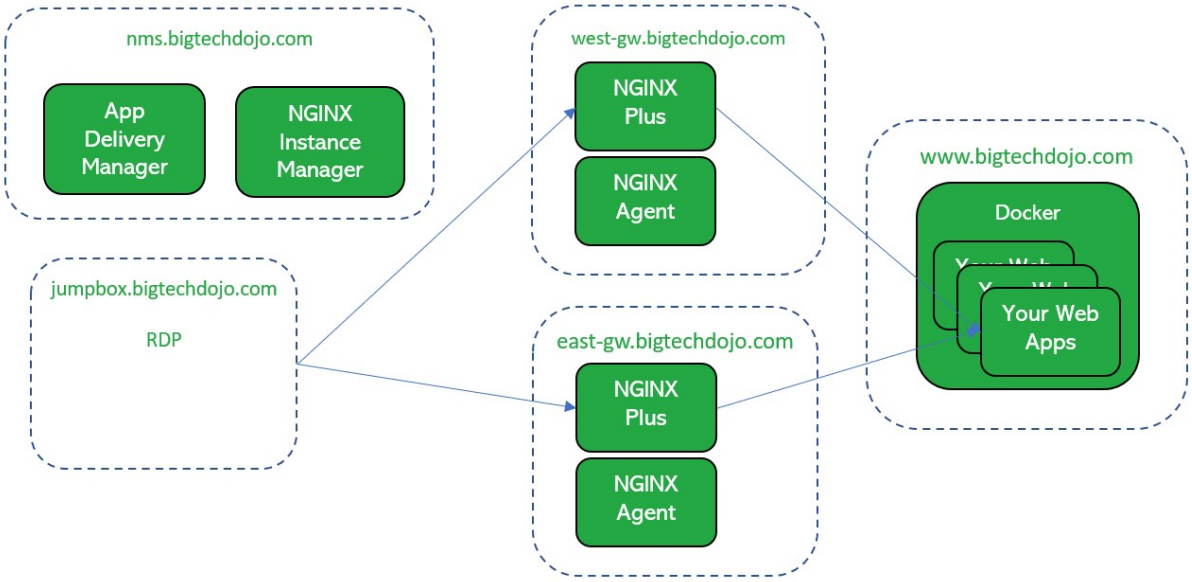
Brewz lives on **www.bigtechdojo.com** server, and consists of multiple microservices a shown here:

```
IMAGE                              PORTS
spa-demo-app_recommendations       0.0.0.0:8001->8001/tcp
spa-demo-app_spa                   0.0.0.0:8081->80/tcp,
spa-demo-app_inventory             0.0.0.0:8002->8002/tcp
spa-demo-app_api                   0.0.0.0:8000->8000/tcp
spa-demo-app_checkout              0.0.0.0:8003->8003/tcp
```

Juiceshop also lives on **www.bigtechdojo.com**, is containerized, but is monolithic, deployed on a single port, but there are two instances of Juiceshop on ports **3000** and **3001** as shown here:

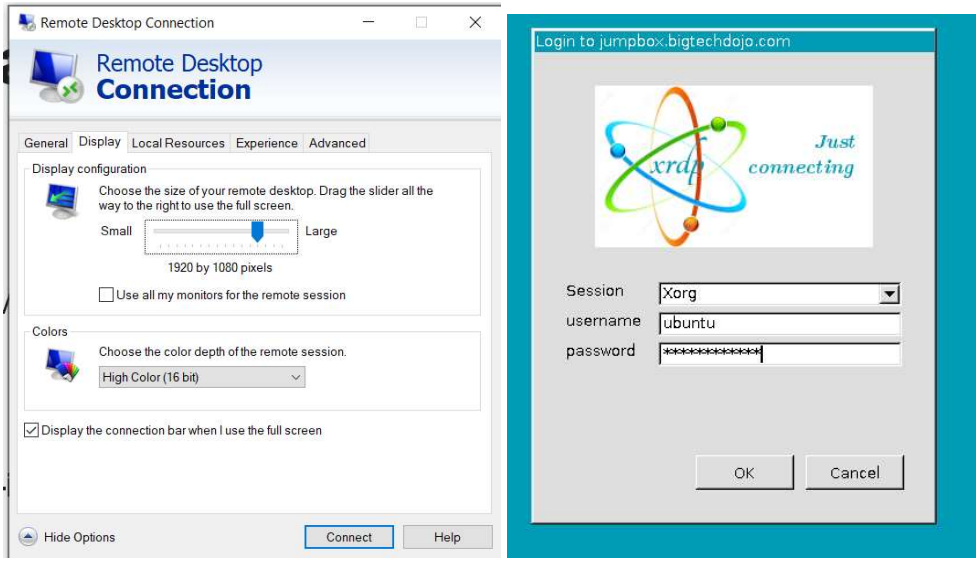
```
bkimminich/juice-shop    0.0.0.0:3001->3000/tcp
bkimminich/juice-shop    0.0.0.0:3000->3000/tcp
```

Diagram



List of Components

1. NMS - this is your NMS and ADM management platform. Creds are shown in the UDF deployment->NMS component->details
2. WEST-GW - this is one of your NGINX+ and NGINX-AGENT instances.
3. EAST-GW - this is another
4. WWW - this is a box of web servers, with one listening on port 80
5. UBUNTU - this is simply a jumpbox. creds are creds are shown in the UDF deployment->Ubuntu component->details.
    1. Access via RDP, and then do all your work from the jumpbox.
    2. This box is faster, smoother, more fun and easier than a Windows Jumpbox. Don't worry it has Firefox on it.
    3. Pro Tip: When you download the RDP file, right-click and edit the file and set color depth to 16-bit, set the screen resolution to slightly less than your monitor, set the username to ubuntu, and enable saving username. Use Ctrl-Shift-V to paste.



Detailed Requirements & Information

- The platform infrastructure team has already configured the software and hardware for you, including NGINX Management Suite, App Delivery Manager, the NGINX instances, and the application servers.

  You are the application delivery team, and in that role, you will consume the platform to deliver the application. The business wants the Juice application deployed in the West, and the Brewz application deployed in the East.

  Apps need to be accessed securely over the Internet at the domains as shown in the diagram.

  You will know you have succeeded when you can browse securely to the West gateway and use the Brewz app, and you can browse securely to the East gateway and use the Juice app.

  You will do everything from the Ubuntu RDP jumpbox

Spoiler Alert – Do not proceed unless you want to see the step-by-step solution

Solution

In the NGINX Management Suite web interface, you access the App Delivery Manager (ADM) features by performing the following operations. You will do everything from the Ubuntu RDP jumpbox.

1. Log into the Ubuntu jumpbox via RDP: credentials can be found in UDF under “Ubuntu JumpBox RDP”, “Details” and then “Documentation” tab
2. Start Firefox on the Ubuntu jumpbox
3. Click on the “West Gateway” book mark link in Firefox, and notice nothing is there. This is because West Gateway is a totally unconfigured nginx instance.
4. Click on the NMS bookmark in the bookmark bar and login. Credentials can be found in UDF under “NMS-ADM”, “Details” and then “Documentation” tab
5. From the Launchpad, select the **NGINX Instance Manager** card to see your instances that have already been deployed for you.
    1. These have been put into two Instance Groups, region-1 and region-2.
    2. Click on one of the instance groups and click “edit config” to see the “blank” configuration.
6. Now, go back to the Launchpad and select the **App Delivery Manager** card

Create an Environment

The first resource you need to create, if one doesn’t already exist, is an Environment resource. This can be accomplished by taking the following steps:

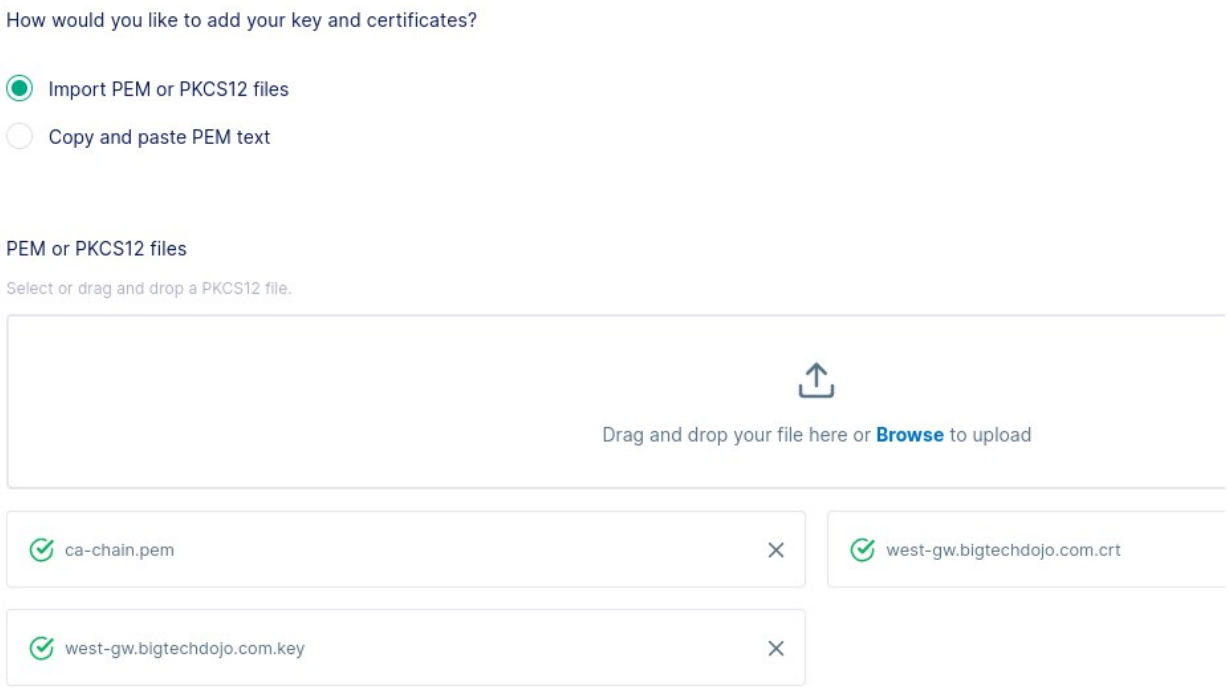
1. Select **Environments** from the **App Delivery Manager** list in the left-hand sidebar. The list of existing environments will then display.
2. Select **Create Environment** on the right-hand side of the list. A panel will appear that allows you to configure the environment.
3. Enter the value **Production** for the **Name** field. This logical environment is not just for one app, but for all your internet-facing websites. You can take the defaults for all the other fields (this exercise does not require customized templates).
4. Select **Submit** to finish creating the environment.

Create a Gateway for Juice Shop

The gateway controls how traffic will route through an NGINX instance to get to the app workloads.

1. Select **Gateways** from the **App Delivery Manager** list in the left-hand sidebar. The list of existing gateways will then display.
2. Select **Create Gateway** on the right-hand side of the list. A panel will appear that allows you to configure the gateway.
3. From the Configuration page of Create Gateway, enter the gateway name as **West Gateway**. You can accept defaults for the next two fields.
4. For the environment field, select the environment **Production** that you previously created.
5. Select **Next** to get to the **Placements** page.
6. The platform team should have created an instance group **region-1**. Select **Add Placement** and from the Instance Group **Refs** dropdown, select **region-1**. Then click **Done**.

7. Select **Next** to get to the **Hostnames** page.
8. Select **Add Hostname** then enter **https://west-gw.bigtechdojo.com** for the Hostname.
9. In the Shared TLS Settings section, select **Create New**
10. In the **Create Certificate** page enter the name as **west-gw.bigtechdojo.com**
11. Select Import **PEM or PKCS12 files** and upload **ca-chain.pem**, **west-gw.bigtechdojo.com.crt** and **west-gw.bigtechdojo.com.key** found on the ubuntu Jumphost under /home/ubuntu



12. Under **Associated Instances** select **west-gw.bigtechdojo.com**
13. Select **Submit** to create the certificate
14. Back to the **Create Gateway** page select **Submit** to complete the Gateway
15. When Gateway deployment is completed, Status will be shown as **Configured**

Create your Application for Juice Shop

Follow these steps to create the applications:

1. Select **Apps** from the **App Delivery Manager** list in the left-hand sidebar. The list of existing apps will then display.
2. Select **Create App** on the right-hand side of the list. A panel will appear that allows you to configure the app.
3. Enter the value **Juice** for the Name field. Select **Production** for the **Environment** field. You can take the defaults for all the other fields.
4. Select **Submit** to finish creating the app.
5. Status will be displayed as **Configured**

Create the Juice Shop Production Web Component

The app we just created is a wrapper that can be composed of multiple components, each potentially referencing a unique service or microservice. To create the production component, perform these steps:

1. You should be on the **Apps Overview** page at this point. Select the app that was just created in the list by clicking the app name.
2. The main display will now show basic metrics for the app. We are not, at this point, interested in the metrics, but from this page we can create a component. At the top of the page, select Web Components.
3. The list of web components will appear, but should be empty.
4. Select **Create Web Component** on the top right-hand side of the display. A panel will appear that allows you to configure the component. There will be several pages of configuration that will need to be performed.
5. On the first page (Configuration), enter the value **Juice** for the Name field.
6. The only other field that needs to be set on this page is the **Gateway Refs** field. Under this field, select **West Gateway**.
7. Click **Next** to advance to the URIs page.
8. Enter **/** for the URI (if you are not able to enter a value, click the pencil icon to edit the URI).
9. Click **Next** to proceed to the Workload Groups page and select **Add Workload Group**
    a. In the **Workload Group Name** field, enter **Juice Servers**
    b. In the Backend Workload URIs section, enter for the URI field and click Done:
        http://www.bigtechdojo.com:3000
    c. Select Add Backend Workload URI to add another workload, and enter:
        http://www.bigtechdojo.com:3001
    d. Click **Done**, then Click **Done** for the overall Workload Groups page.
10. Select the **Submit** button to complete the component configuration.
11. When the configuration is applied **Status** will be shown as **Configured**

Testing

1. Select the **West-gw** bookmark, or refresh the tab if you had one open, and you will see the application.
2. In NMS, navigate to Instance Manager, go to Instance group **region-1**, click on “Edit Config” and you will see the configuration that ADM has created.

Create a Gateway for Brewz

Follow the steps as shown above in the “Create a gateway for Juice Shop” section, with the following changes:

1. Configuration: name the gateway **East Gateway**
2. Placements: Place this gateway in **region-2** of the country, for low latency.
3. Hostnames: Enter **https://east-gw.bigtechdojo.com** for the **Hostname**
4. Create a certificate using **ca-chain.pem**, **east-gw.bigtechdojo.com.crt**, and **east-gw.bigtechdojo.com.key**
5. Click **Submit**

Create your Application for Brewz

Now that you have the hang of a simple monolithic application, you are going to deploy the microservices application Brewz. Brewz lives on the following ports:

```
IMAGE                          PORTS
spa-demo-app_recommendations   0.0.0.0:8001->8001/tcp
spa-demo-app_spa               0.0.0.0:8081->80/tcp,
spa-demo-app_inventory         0.0.0.0:8002->8002/tcp
spa-demo-app_api               0.0.0.0:8000->8000/tcp
spa-demo-app_checkout          0.0.0.0:8003->8003/tcp
```

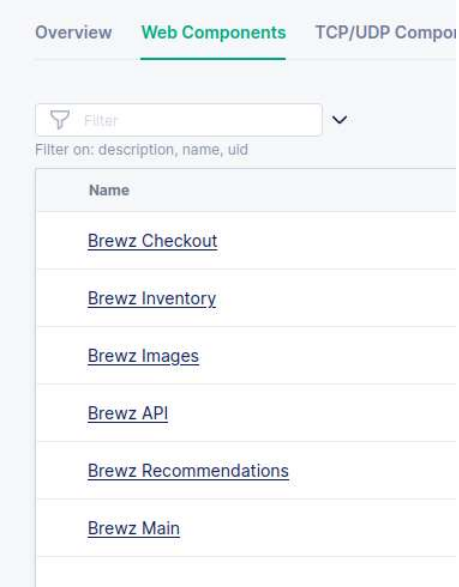
1. Create a new application
2. Enter **Brewz** for the **Name** field and select **Production** in the **Environment** field

The front-end of the Single-Page Application directs traffic to each of these containers based on the following routes:

URI	Port	Web Component
/checkout	8003	Brewz Checkout
/	8081	Brewz Main
/recommendations	8001	Brewz Recommendations
/inventory	8002	Brewz Inventory
/api	8000	Brewz API
/images	8000	Brewz Images

1. Create each of the 6 Brewz Web Components

You are going to create a web component for each Brewz microservice so that you end up with the following 6 web components:



The detailed steps are provided for the **/checkout** route, you are going to learn from these steps and create the other routes.

Follow the steps you did above for the Juice app, with the following changes:

12. On the first page (Configuration), enter the value **Brewz Checkout** for the **Name** field.
13. The only other field that needs to be set on this page is the **Gateway Refs** field. Under this field, select **East Gateway**. Then click **Next** to advance to the URIs page.
14. Enter **/checkout** for the URI (click the pencil icon to edit the URI).
15. Click **Done** to save the URI
16. Click **Next** to proceed to the Workload Groups page.
    a. In the **Workload Group Name** field, enter **Brewz Checkout**
    b. In the Backend Workload URIs section, enter for the URI field:

http://www.bigtechdojo.com:8003

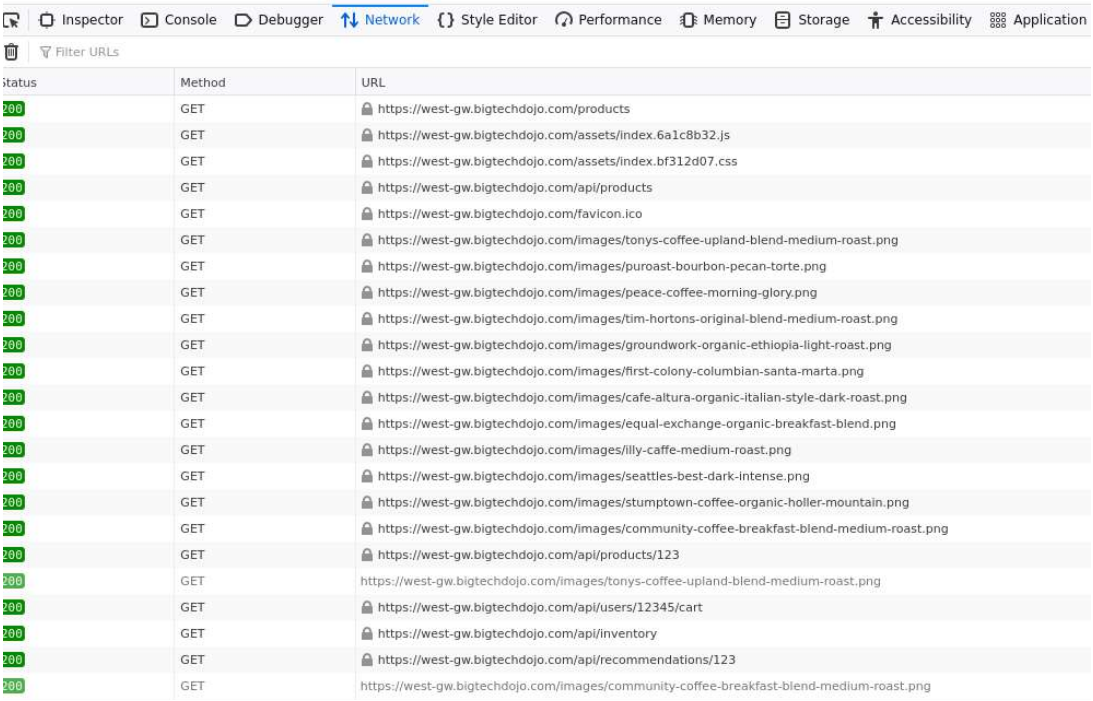
c. Click **Done**, then click **Done** for the overall Workload Groups page.

17. Select the **Submit** button to complete the component configuration.
18. Create the remaining five routes to Brewz microservices as per the table above. Make sure the Backend Workload URI port is configured based on the table.

Testing

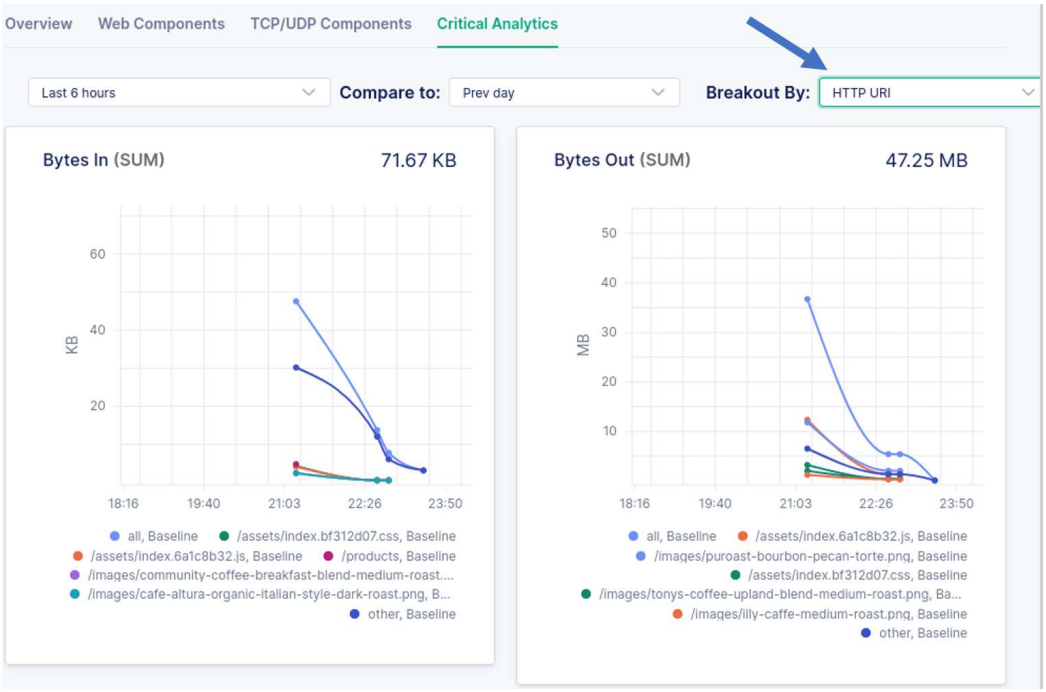
1. Select the **West-gw** bookmark, or refresh the tab if you had one open, and you will see the application.
2. In NMS, navigate to Instance Manager, go to Instance group **region-1**, click on “Edit Config” and you will see the configuration that ADM has created.

View Paths in Network Inspector



Critical Analytics

In the ADM module select **Apps** then **Juice** and open the **Critical Analytics** tab. **Select** Breakout By: HTTP URI and select **Last 30 minutes**


Overview
Web Components
TCP/UDP Components
Critical Analytics
Last 6 hours
Compare to:
Prev day
Breakout By:
HTTP URI
Bytes In (SUM)
71.67 KB
KB
60
40
20
18:16
19:40
21:03
22:26
23:50
all, Baseline
/assets/index.bf312d07.css, Baseline
/assets/index.6a1c8b32.js, Baseline
/products, Baseline
/images/community-coffee-breakfast-blend-medium-roast....
/images/cafe-altura-organic-italian-style-dark-roast.png, B...
other, Baseline
Bytes Out (SUM)
47.25 MB
MB
50
40
30
20
10
18:16
19:40
21:03
22:26
23:50
all, Baseline
/assets/index.6a1c8b32.js, Baseline
/images/puroast-bourbon-pecan-torte.png, Baseline
/assets/index.bf312d07.css, Baseline
/images/tonys-coffee-upland-blend-medium-roast.png, Ba...
/images/illy-caffe-medium-roast.png, Baseline
other, Baseline

Module 2 – NAP WAF

In this module, you are going to enable NAP WAF.

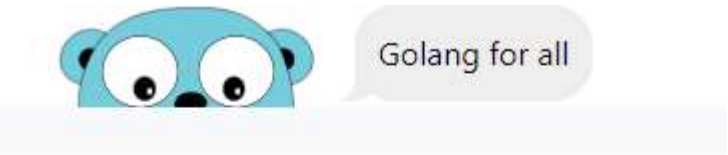
By selecting the NAP Policy in the Template, the pre-complied policy TGZ is automatically pushed out from the ADM host to the NGINX Dataplane host.

Furthermore, if the app developer clears any policy from the WAF Template, the agent removes the TGZ from the dataplane host - very impressive

Enabling Templates

1. Edit your **Production** environment (click on the three dots) and select **Use Cases** templates and select the following templates: **WAF**
2. Click on your Juice App (underlined)
3. Edit your Juice Web Component
4. Go into custom extensions, select the Nginx App Protect default policy
5. Submit
6. Review the resulting config in Instance Manager on the region-1 instance group.

Module 3 – More Templates



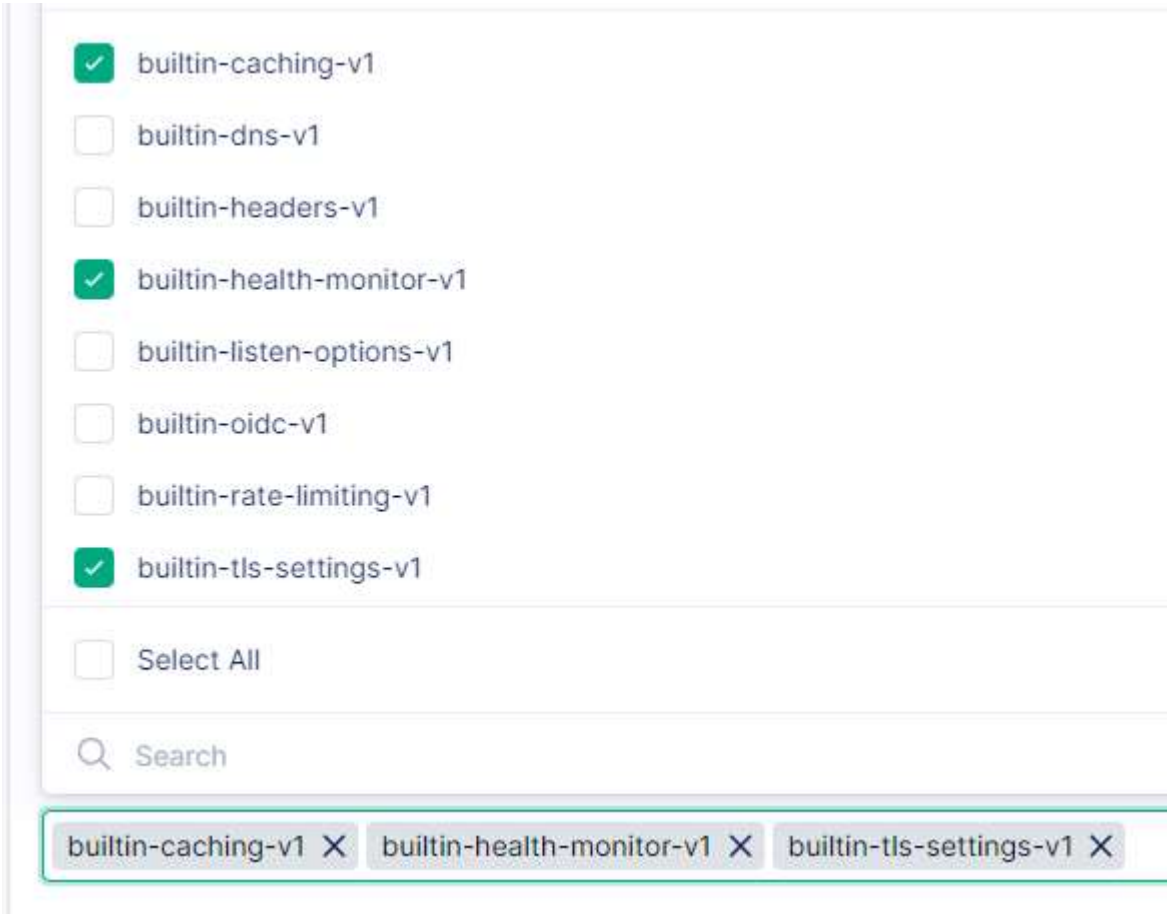
Templates allow a customer to extend ADM.

In this module, you are going to enable existing templates. After that, you will install new templates into ADM that professional services has created to enable remote syslog logging to a rsyslog server.

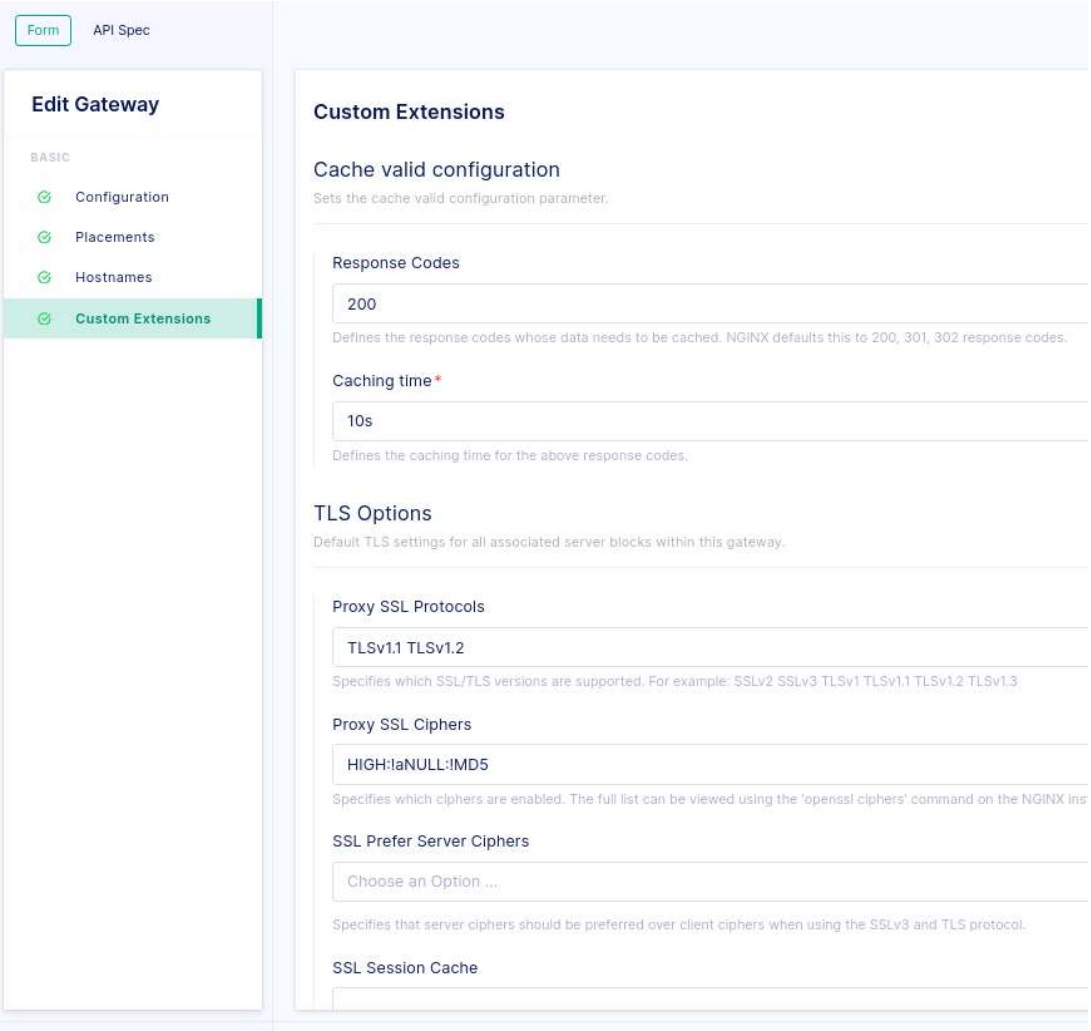
Enabling Templates

You are going to enable the Caching, Health Monitoring, and TLS templates.

1. Edit your **Production** environment (click on the three dots) and select **Use Cases** templates and select the following templates:

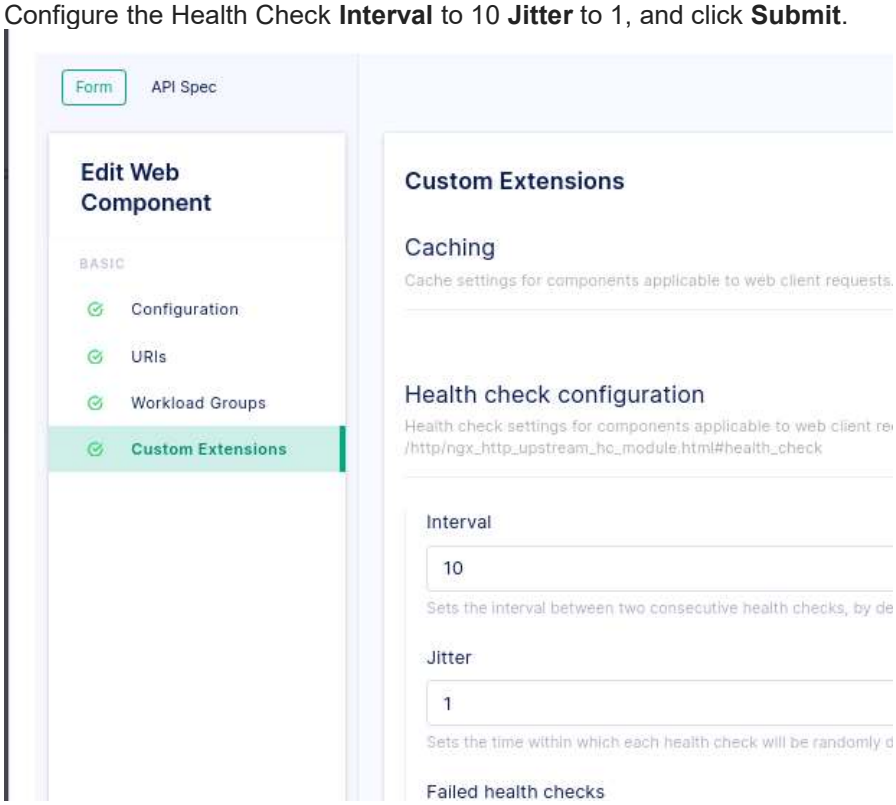


2. From the **Gateways** menu **Edit** your **West Gateway** and select **Custom Extensions** and enable the following Custom settings:
    1. Caching
        1. Response Codes: **200** and Caching Time: **10s**
    2. TLS Options
        1. Proxy SSL Protocols: **TLSv1.1 TLSv1.2**
        2. Proxy SSL Ciphers : **HIGH:!aNULL:!MD5**



3. Click **Submit**
4. From the **Apps** menu select **Juice** and the **Web components** tab. Edit your **Juice Shop** web component by clicking on the three dots and select **Custom Extension**

5. Configure the Health Check **Interval** to 10 **Jitter** to 1, and click **Submit**.



6. View the resulting configuration in Instance Manager by selecting the **Instance Groups** section and opening **region-1**
   Look for **health_check** to see configuration lines added by the template.

```
    # Created by Gateway: West GW(d2d0b917-2d84-4dcf-9b15-3c43ab1d4a78)
    server {
        server_name west-gw.bigtechdojo.com;
        listen 443 ssl reuseport;
        proxy_cache_valid 200 10s;
        # Usecase: builtin-server-tls-v1
        # Template: server-gateway.tmpl
        # Gateway name: West GW
        proxy_ssl_protocols TLSv1.1 TLSv1.2;
        proxy_ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_certificate /etc/nginx/aux/west-gw.bigtechdojo.com.crt;
        ssl_certificate_key /etc/nginx/aux/west-gw.bigtechdojo.com.key;
        status_zone d2d0b917-2d84-4dcf-9b15-3c43ab1d4a78;
        f5_metrics_marker environment 18af103e-fda0-4470-af81-1b4c320c4bc2;
        f5_metrics_marker gateway d2d0b917-2d84-4dcf-9b15-3c43ab1d4a78;
        location = /_health_check_b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4 {
            internal;
            proxy_set_header Host $host;
            health_check jitter=1 interval=10 fails=1 passes=1 uri=/;
            proxy_pass http://b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4;
        }
        # Generated by web component jjj(b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4)
        location / {
            status_zone b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header Host $host;
            proxy_set_header Connection "";
            proxy_http_version 1.1;
            proxy_pass http://b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4;
            # metrics
            f5_metrics_marker app a5391940-57d4-480f-ba55-2c6cb91594fa;
            f5_metrics_marker component b9ca5f77-0b84-4df4-9ad9-6e6ac0cdd6f4;
        }
```

Installing Custom Templates

Here you are in the role of the SE **helping** the **Platform Team**, so you are going to touch the NMS at the Operating System level. So you have changed hats for this part.

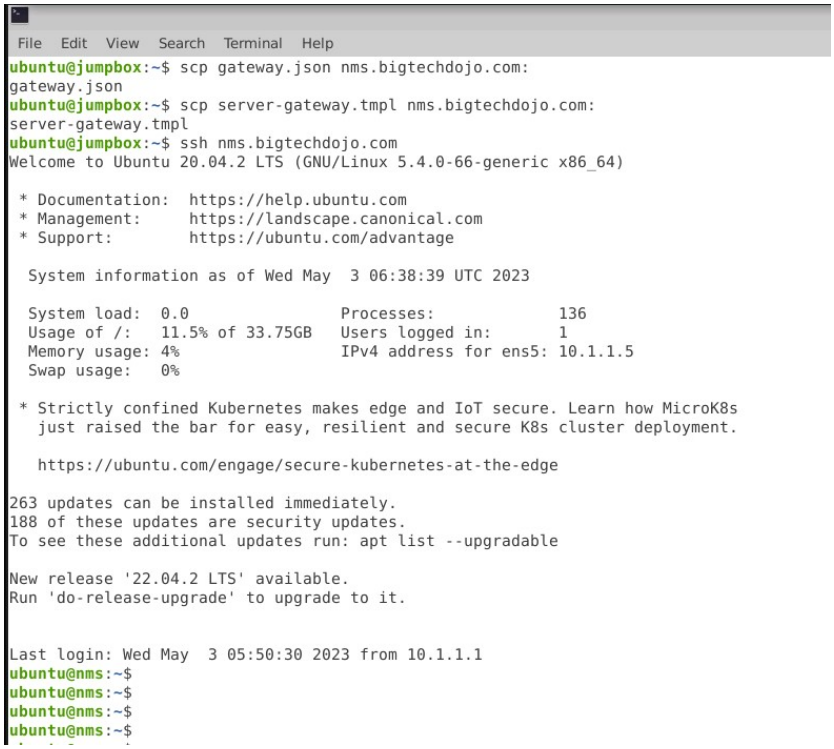
Imagine that your customer has a requirement for a feature that is not included in the product, but can be enabled with Templates.

In this example, the templates have already created by F5 professional services or F5 SA, and are located on your jumpbox in the /home/ubuntu directory.

You just need to help the customer Platform Team install them and configure them in ADM.

1. Open a terminal window on the jumpbox
2. In this terminal window, SCP copy the **gateway.json** and **server-gateway.tmpl** to the NMS system, and then SSH to the NMS system (ssh keys are already set up) (you can copy and paste (On a Windows system, use CTRL-SHIFT-V to paste into the RDP window)

```
scp gateway.json nms.bigtechdojo.com:
scp server-gateway.tmpl nms.bigtechdojo.com:
```



3. SSH to the NMS system by running **ssh nms.bigtechdojo.com** in the terminal window:

```
ssh nms.bigtechdojo.com
```

4. Enter exactly the following commands **one-by-one** on the NMS system (On a Windows system, use CTRL-SHIFT-V to paste then **one-by-one** into the RDP window. Do not copy all the lines and paste them at once, you can copy and paste them **one-by-one**)

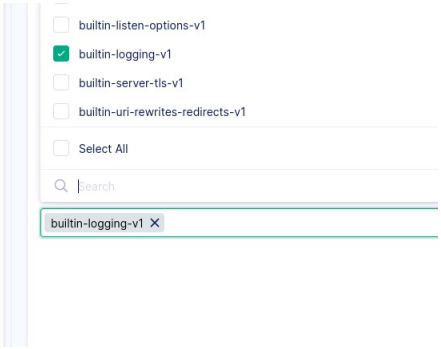
```
sudo bash
cd /etc/nms/modules/adm/templates/usecases
mkdir builtin-logging-v1
cd builtin-logging-v1
cp /home/ubuntu/gateway.json .
cp /home/ubuntu/server-gateway.tmpl .
chown nms *
chgrp nms *
ls -altr
exit
exit
```

5. That is the end of the work for the Platform team.
6. In your jumpbox Terminal window, fully exit out of your SSH session to NMS by typing exit so you are back to your jumpbox prompt.
7. In your jumpbox terminal window, type

```
tail -f /var/log/syslog
```

8. Send web traffic to your gateway by refreshing the browser, and notice you do not yet see HTTP requests in your syslog on the jumpbox

9. Now, go back to your **Browser** window. In the ADM section go to your **Production** environment, select **Edit** from the "..." link on the right side of the screen. Click on the **Templates** section and select your **builtin-logging-v1** template in the **Use Cases** section, then **Submit** all changes

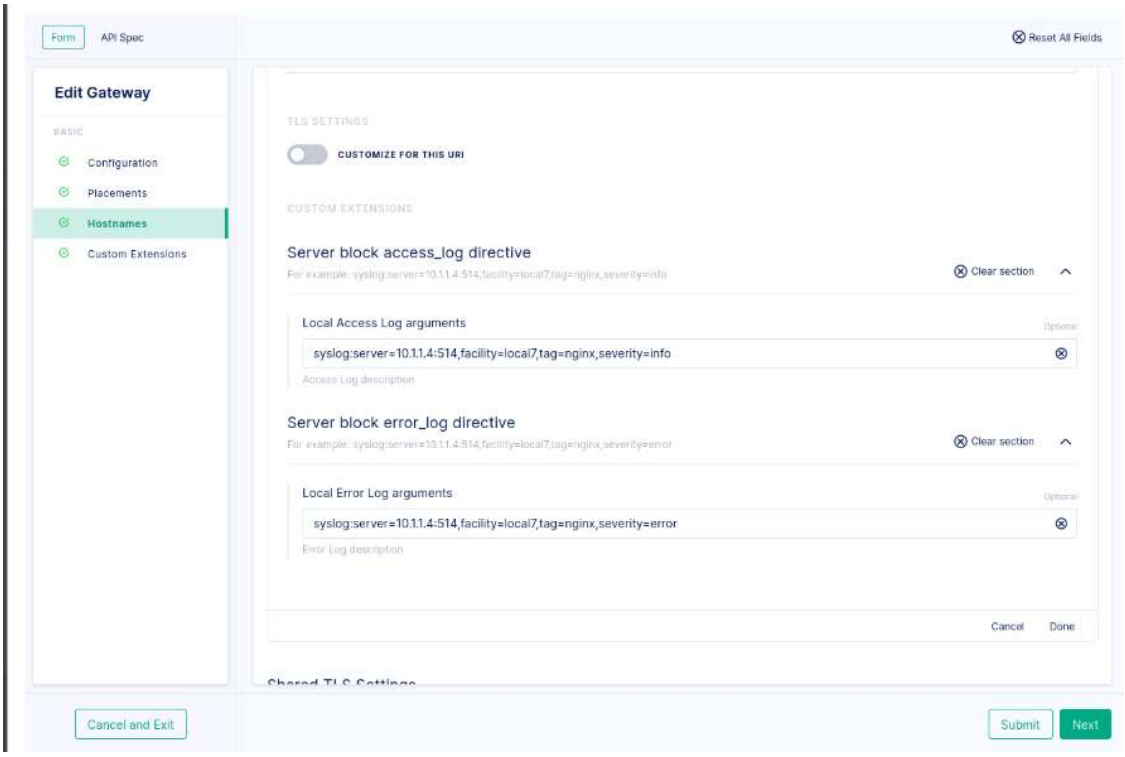


10. Edit your **West Gateway**
11. Select the **edit** button to modify the **Hostname**.
12. You will see new **Server block access_log** and **Server** block **error_log** fields, from the template.
13. Enter the following for the **Access Log** field:

```
syslog:server=jumpbox.bigtechdojo.com:514,facility=local7,tag=nginx,severity=info
```

14. Enter the following for the **Error Log** field:

```
syslog:server=jumpbox.bigtechdojo.com:514,facility=local7,tag=nginx,severity=error
```



15. Click **Submit**

16. View the resulting configuration in Instance Manager, you will see your **access_log and error_log** directives created under the Server block.

17. On your jumpbox, tail -f /var/log/syslog, send web traffic to your gateway by refreshing the browser, and make sure you see nginx access log entries in your syslog