Creating an EFT™ HA Cluster in Amazon Web Services (AWS)

GLOBALSCAPE

GlobalSCAPE, Inc. (GSB)	
	Corporate Headquarters
Address:	4500 Lockhill-Selma Road, Suite 150, San Antonio, TX (USA) 78249
Sales:	(210) 308-8267
Sales (Toll Free):	(800) 290-5054
Technical Support:	(210) 366-3993

Web Support: http://www.globalscape.com/support/

April 17, 2018

Disclaimer

The information contained within this document, including instruction steps, techniques, best practices, recommendations, screenshots, sample source code, etc. is provided "AS IS." GlobalSCAPE, Inc. disclaims all warranties, expressed or implied, including and without limitation, the warranties of merchantability and of fitness for any purpose. GlobalSCAPE, Inc. assumes no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of this information or the techniques provided, even if advised of the possibility of such damage. This information is provided for educational purposes only, and is not formally supported, maintained, or otherwise endorsed by GlobalSCAPE, Inc.

Contents

Introduction

Audience

This document is intended for technical personnel who have made a decision to deploy EFT Enterprise as a static (non-auto scaling), Highly Available (HA), Active-Active cluster in Amazon Web Services (AWS).

Please refer to companion documentation on choosing between cloud deployment options if you are looking for guidance on the various cloud deployment options available, including pros and cons of each in terms of reliability, scalability, redundancy, and cost factors.

Document Goals

The goal of this document is to offer *guidance* (see Disclaimer), in the form of step-by-step instructions, for deploying EFT in HA mode on Amazon's public could infrastructure, known as Amazon Web Services (AWS). When configured correctly, an HA cluster can provide a resilient, geographically redundant setup that can in turn deliver enterprise-level Managed File Transfer (MFT) services under the most demanding, mission-critical situations.

Document Scope

Amazon AWS specific deployment of a two-node active-active MFT cluster that leverages a number of AWS infrastructure services, Globalscape's manage file transfer application, EFT Enterprise, and Samba, to achieve a highly available, geographically disparate, HA cluster.

This setup guide does not provide a detailed primer on AWS infrastructure management, fundamental networking concepts, auto-scaling clusters, or hybrid, static, auto-scaling clusters.

Assumptions

This document assumes you have:

- A technical background (networking, DevOps, or datacenter management) and at least a rudimentary knowledge of cloud infrastructure management.
- Chosen to deploy EFT Enterprise in a static cluster in an active-active configuration behind AWS Elastic Load Balancing (ELB) services.
- Access to EFT activation keys (unless deploying a trial setup), and access to an AWS account
- Authority over service subscriptions and users, and authority and ability to license third-party software, as described under the pre-requisites section below.
- An understanding of the business logic they want to accomplish with EFT.
- At least somewhat familiar knowledge of EFT, enough to configure an EFT Site, add users, setup auditing and retrieve reports, and setup event rules.

Constraints

Neither Amazon's Elastic Block Storage (EBS) nor Elastic File Share (EFS) are suitable direct storage mechanisms for EFT running in HA mode. EBS volumes can only be mounted to a single instance at a time and are tied to a single Availability Zone (thus suitable for local storage only) and EFS with Microsoft Windows Amazon EC2 instances is not supported by Amazon. EFS, however, can be used to back a GNU/Linux host running Samba, which EFT can use as its storage backend via the SMB protocol.

Risks and Risk Mitigation

Cost

AWS charges based on infrastructure usage for everything from compute (EC2), networking, storage (EFS), notification (SNS), queueing services (SQS), and more. If not tightly controlled and monitored, it is possible to rack up huge monthly expenses related to AWS services.

You should make every effort to understand the effects of choosing various infrastructure options, such as the difference between dynamically allocated vs. dedicated IOPS, the difference between a lower tier "T" type burstable compute instance and a high end "M" type memory intensive compute instance. Another way to avoid surprises is to setup billing and usage monitoring services and then configure AWS's CloudWatch services to trigger notification alarms or even shutdown services that exceed a given threshold. Both of these risk mitigation techniques are your responsibility.

Availability

The proposed architecture attempts to balance redundancy and cost. As such, a single point of failure risk is introduced by nature of having a single Linux node for the Samba file share. Furthermore, if you fail to properly follow instructions and deploy cluster nodes in the same subnet or availability zone (AZ), then a risk is introduced in that if that zone were to experience problems, the entire cluster would be affected. Finally, deploying a two-node cluster introduces a single-point-of-failure risk should one node be brought down for maintenance, as only one node will remain available to handle traffic during that timeframe.

Availability risks can be minimized by creating an auto-recovery CloudWatch event for the Samba file share. Note however that this process is outside the scope of this document.

Carefully following the instructions can help prevent setting up nodes in the same AZ, and deploying at least a three-node cluster (with each node in a separate AZ), can help mitigate the effects of planned downtime due to node maintenance. Keep in mind however that even if deployed properly across multiple AZs, a container of AZs (region) can still be a single point of failure, as is evidenced by the S3 east-coast outage (which affected an entire AWS region and contains a number of AZs) experienced by Amazon in February of 2017. Note that while theoretically possible, we currently do not promote or support inter-region clusters.

Security

Security risks include but are not limited to:

- Improper security group configuration, especially for RDP services. Mitigation: Limit RDP access to your IP, and use complex passwords and non-standard names for administrative access.
- Public access to exposed ports such as 22 and 443 may result in probing for weaknesses for applications listening on those ports. Proper configuration of EFT for DoS/Flood prevention, account login failure handling, and password complexity should help mitigate this risk.
- Amazon's Elastic Load Balancing (ELB) service requires the application's private certificate to achieve session stickiness for HTTPS traffic. This risk to privacy is likely small, but is present and should be noted.

Business

There are intrinsic business, security, and privacy risks that comes with trusting a third party with your infrastructure. That topic is vast, well covered in the industry, and outside the scope of this document.

There is a business continuity and financial risk in trusting a single third-party vendor due to vendor lock-in. For example, what if the IaaS provider went out of business or suffered catastrophic outages? Or what if the vendor decides to arbitrarily raise prices? One way to mitigate this risk is to consider deploying your solution in a redundant fashion across multiple IaaS providers, or a hybrid deployment in which a redundant DR site is maintained, even if offline, with the ability to quickly switchover services to the alternate vendor or on-premises, should continuity or financial risks occur.

Architecture

Overview

This document proposes a simple two-node architecture with each node in different availability zones (AZs), which are physically separate data centers connected by high-speed fiber within the same region.

An EFT server is deployed in each AZ on Windows Server 2016 Amazon Machine Images (AMIs), along with a single Linux AMI in one of the zones. The Linux node will be setup as a Samba share. NFS will be used to back the Samba server to provide a single, highly available, and redundant storage solution. Finally, this architecture leverages a number of Amazon services to facilitate load balancing and node synchronization of configuration changes, and covers a few optional services for things like auditing and directory services.

Diagram

Figure 1 outlines a diagram of a two-node cluster configuration. As mentioned under the Risks section, this is not the ideal three-node-or-more cluster scenario, nor does it show any optional services like RDS or ADS. The reason for these shortcomings was to keep the diagram and architecture simple enough for quick deployment, while providing a solid foundation that you can build upon to produce a more robust solution as desired.

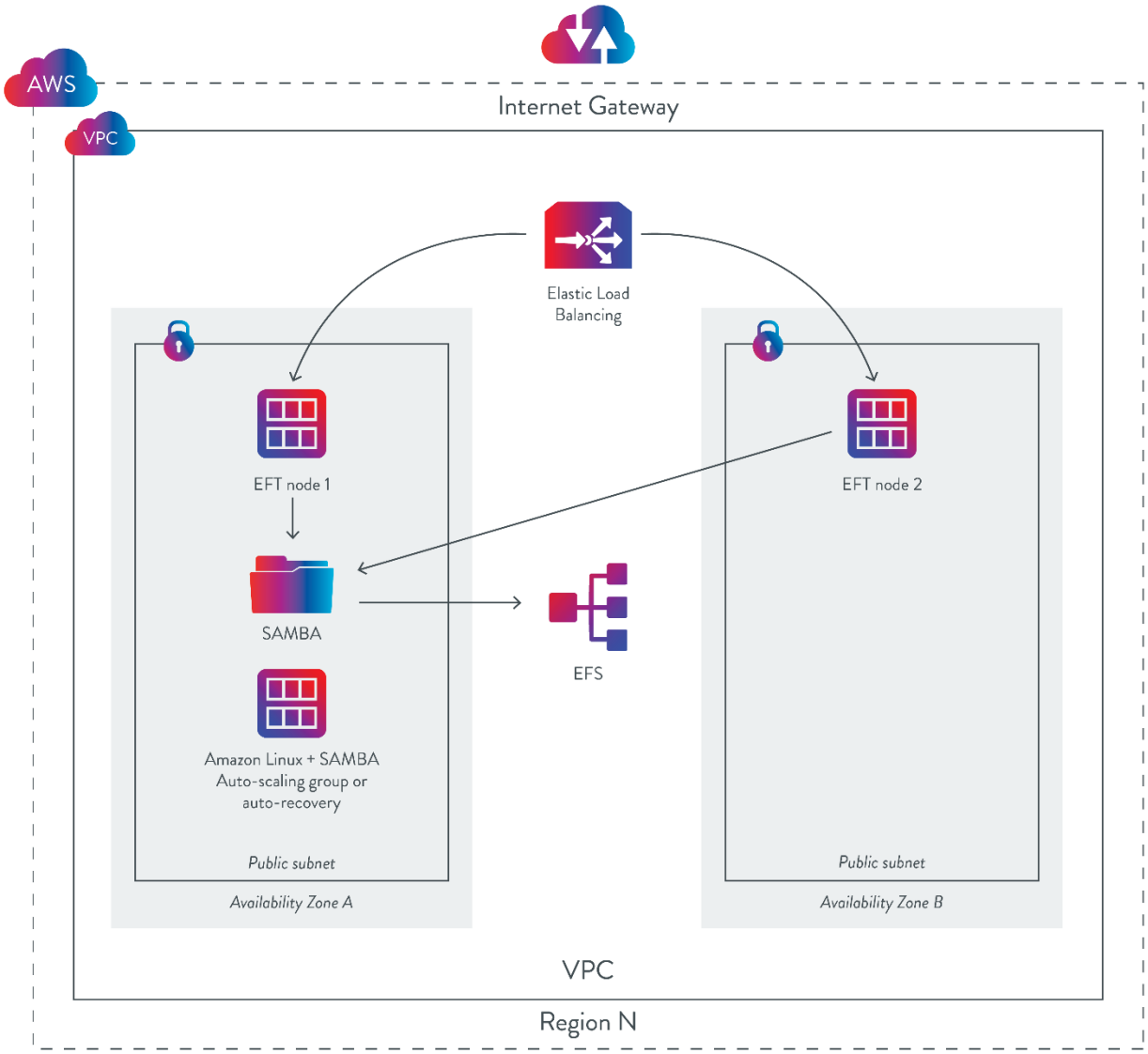


Figure 1 - Two Node EFT Cluster

Future Considerations

Globalscape's consulting and professional services team can assist with more robust and advanced deployment configurations, including:

- Multi-node clusters
- Adding DMZ Gateway® to clusters
- Mixing static and auto-scaling clusters
- Assist in the creation of AWS command-line interface CLI code, LAMBDAs, and cloud formation templates (CLTs) to help automate and reduce the level of effort required to deploy EFT in an HA cluster

Cost Calculator

Overview

Before proceeding, you should understand the cost involved in the various components necessary to deploy EFT in an HA cluster on AWS. While this seems straight forward, determining the cost of a cloud-based deployment is actually simple task. There are a number factors that can determine the cost, such as the size of the compute instances, the amount of traffic expected, the storage requirements, the specific modules purchased for EFT, the number of nodes in the cluster, and so on. And, while it is not within the scope of this document to compare on-premises to cloud deployments, we can provide a few tools that you can leverage to at least get a rough idea as to the costs involved.

AWS Costs

The tool at this link https://calculator.s3.amazonaws.com/index.html can be used to estimate the cost of most AWS components.

Prerequisites

Please note AWS infrastructure services require a billing account or credit card. Notwithstanding AWS's free trial period for new AWS customers (which restricts instance sizes amongst other limitations), AWS will begin billing you immediately for services used. In contrast, the EFT application can be deployed in trial mode, which allows for full functionality for thirty days. This allows you to evaluate EFT without being charged. Once the trial is over you can either request a trial extension or purchase a full license if desired.

The following prerequisites are needed to create the cluster:

AWS

1. An active AWS account (subscription) for billing purposes.
2. Sufficient privileges to configure and deploy AWS services.

3. Authority to create IAM users and assign security policies.
4. Authority and capability to pay for AWS services rendered.

Globalscape

1. EFT **installer**.
2. Your **EFT license** activation key (unless using trial version).
3. Your **EFT modules license** activation key(s).

HA Cluster Setup

Overview

The following sub-sections will take you through the process of setting up an HA cluster from start to finish. The process will follow these steps:

1. Log into AWS
2. Specify the desired Region
3. (Optional) Setup RDS
4. Create a keypair for login access to EC2 instances
5. Create a set of Security Groups
6. Create an EFS volume
7. Launch a Linux instance
8. Setup Samba, EFS, and S3 on the Linux instance
9. Launch a Windows instance
10. Deploy EFT on Windows instance
11. Configure EFT
12. Setup AWS's load balancing services
13. Verify the solution

Log into AWS

Using your AWS login credentials, visit the AWS console, which will either be the main AWS portal:

https://console.aws.amazon.com/console/home

Or a specific IAM portal that has been created for you:

https://[account-number].signin.aws.amazon.com/console

Region Selection

For this exercise, make sure you choose and stick with the same region. In the example below, we chose region 1 (North Virginia - East). It is also key that you place any new services added into the same Virtual Private Cloud (VPC). While the EFT instances will be in the same region and VPC, they will be placed in different availability zones (AZs) or subnets, thus eliminating the AZ as a single point of failure. To get started, we will create a new IAM user that will have the responsibility of creating and accessing the various AWS services.

AWS RDS

This is an *optional* step should you want to use Amazon's Relational Database Services (RDS) for a Database as a Service (using SQL) for EFT Auditing and Reporting (ARM), which requires an ARM activation key. Optionally you can setup a standalone SQL Server (extra cost and no redundancy) or forgo auditing altogether.

1. In the AWS console, navigate to Services > IAM
2. In the RDS Dashboard, click on Launch a DB instance.
3. Choose Microsoft® SQL Server from the list. Do not choose MySQL or any other.
4. Select Express or Standard Edition SE
5. Choose from a production, Multi-AZ for a more resilient but expensive deployment, or choose the Dev Test option, which is a standalone deployment on less powerful hardware.
6. Configure the various *options* including License Model, Engine Version (2016 13.00.216… in this example), instance size (will affect pricing), multi-AZ deployment, and allocated storage.
7. Specify an instance identifier, such as eftdb, along with a username and password.
8. Click **Next.**
9. Choose the same VPC in which you created the topic.
10. Use the default subnet or specify a different one.
11. Choose **No** for whether the instance should be Publicly Accessible.
12. Leave the AZ at the *default* or change if desired.
13. Leave AD Directory set to *none*.
14. Leave all **Database Options** set to their defaults.
15. Enable or disable Backup, Monitoring, and Maintenance as you see fit.
16. Click **Launch DB** instance.
17. From the instance dashboard, select the instance details and record the following:

SRS endpdoint: Reftdb.ccsmvm0w6g3x.us-east-1.rds.amazonaws.com

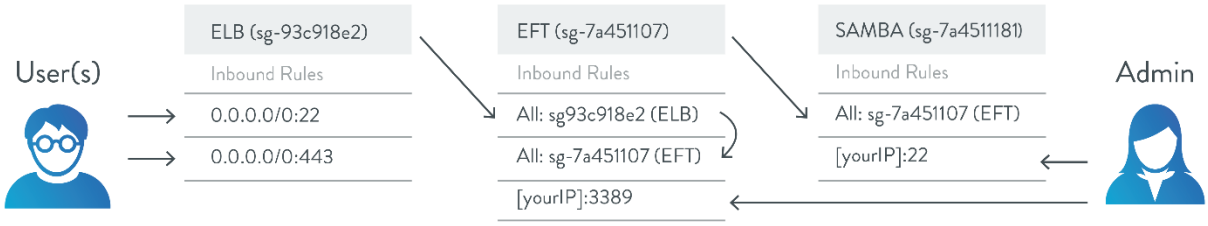
Database name: EFTDB

Username: eftuser

Password: Eft1user!!

AWS Security Groups

Security Groups (SGs) are required to allow the load balancer, Windows, and Linux nodes to communicate with each other. Although it is possible to use a single security group, this tutorial leverages **three** security groups to accomplish our goals. Also note that the SG layout will change if leveraging EFT's DMZ Gateway (see last section in this document).



Load Balancer Security Group

1. In the AWS console, navigate to **Services** > **EC2**
2. Click on **Security Groups** in the left pane
3. Click **Create Security Group**
4. Name the group "**ELBSecGroup**" or similar
5. Add the following inbound rules:
    a. HTTPS, TCP, 443, 0.0.0.0/0
    b. SSH, TCP, 22, 0.0.0.0/0
6. Click **Save**.

Windows (EFT) Nodes Security Group

1. Create another **Security Group** and name it "**EFTSecGroup**" or similar.
2. Add the following inbound rules:
    a. RDP, TCP, 3389, [YOUR_IP] (or 0.0.0.0/0 much less secure!)
    b. All Traffic, All, All, sg-[LBSecGroupID]
    c. All Traffic, All, All, sg-[EFTSecGroupID] (e.g., itself! You may have to add this after creation of the sg.)
3. Click "**Save**"

Linux Node Security Group

1. Create another **Security Group** and name it "**LinuxSecGroup**" or similar.
2. Add the following inbound rules:
    a. SSH, TCP, 22, [YOUR_IP] (or 0.0.0.0/0, but much less secure!)
    b. All Traffic, All, All, sg-[EFTSecGroupID]
    c. All Traffic, All, All, sg-[SambaSecGroupId] (e.g., itself! You may have to add this after creation of the sg.)

3. Click "Save"

The groups allow users to connect to the ELB, which in turn can communicate with EFT, which in turn can communicate with its counterpart nodes (that's why it allows its own security group to communicate with it) and the Samba share.

Write down or remember the name of the security groups:

Security Group 1: ELBSecGroup
Security Group 2: EFTSecGroup
Security Group 3: LinuxSecGroup

EFS Volume

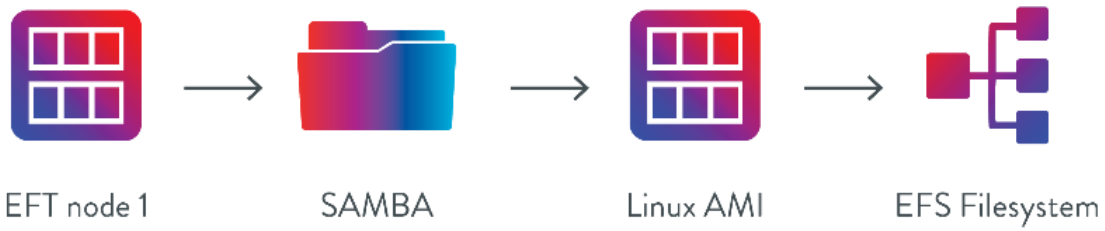
Now we'll set up an EFS volume to mount to our Linux server. This will serve as our data storage backend. While the Linux/Samba server remains a single point-of-failure in this evaluation design, EFS provides a highly available and highly scalable storage solution. To remove the single point of failure represented by a single Samba server, explore auto-recovery, single-node auto-scaling groups (one common way of implementing auto-recovery on AWS), or even Samba clustering or more Windows-based storage backends, such as clustered shared volumes (csv) or Storage Spaces Direct (s2d).

For the purposes of this evaluation deployment, a single Samba server using EFS is more than sufficient, but is not recommended as a production design.

1. Navigate to the EFS control panel in your selected region.
2. Click **Create File System**.
3. Select the VPC in which you're creating your instances.
4. For each configurable availability zone, you can alter the security groups. Add the security group you created for the Linux image to each configurable AZ, and then click **Next Step**.
5. You can add tags on the next screen. You should use a descriptive name such as "eft-ha-eval-efs."
6. You can select between General Purpose and Max I/O performance modes. General purpose is strongly recommended for this evaluation purposes. Please review current costs for EFS prior to selecting Max I/O. It is the user's responsibility to understand the costs they may incur during this procedure.
7. You can choose to enable encryption at rest, but we won't choose this option for our evaluations. Click **Next Step**.
8. You'll have a chance to review your configuration. If everything looks **OK**, click **Create file System**.
9. Record the DNS name under the File system access information. Note that you do not need the Mount Target endpoints; these are maintained for backwards compatibility with systems that used EFS prior to implementation of the single DNS name endpoint.

AWS Linux

This next step sets up a single instance of an Amazon Linux AMI that will host the Samba share. EFT will write to the share, which ultimately writes to an EFS mount.



1. **Find a Linux AMI in the AWS marketplace, such as ami-c58c1dd3 (for us-east-1)** Amazon Linux AMI (HVM / 64-bit). (Note that AMIs are updated frequently, and the example AMI id is unlikely to be valid in the future. Please use the latest AMI available for your region. Also note that these instructions are specific to Amazon Linux, and may differ slightly if you choose a different distribution.)
2. For the Security Group, choose the one you created earlier for the Linux instance or specify a new one, or use the launch wizard default (which opens port 22). Just make sure you add the **EFTSecGroup** security group to a rule (All Traffic, TCP, ALL, EFTHA) such that the EFT nodes can talk to this node.
3. Note that this documentation assumes familiarity with EC2. If you do not have an EC2 keypair set up, you will need to do so in order to finish launching this instance. If you need information on how to proceed, review Amazon's AWS EC2 Keypair documentation at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html
4. Record the private IP of the Linux instance, as you will need it later. For example, File Share IP: "172.31.1.170"
5. SSH into Linux using Putty or similar. Guide here: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html

Samba

A Samba share and authorized user will be configured next.

1. After you SSH into the Linux box, run the following command:

   ```
   $sudo yum install Samba dhcp amazon-efs-utils
   ```

2. Enter a username and password of your choice. Below is an example:

   ```
   Username for account: "eft-user"
   Password for account: "Eft1user!!"
   ```

3. Enter a desired mount point of your choice. Below is an example:

   ```
   Mount Point: "/mnt/eft-share"
   ```

4. Mount EFS Volume to the mount point. You can find detailed instructions on how to mount your mount point in the EFS console. Just view the details for your EFS volume and click Amazon EC2 mount instructions (link). This may include installing software. For example, you may need to:
   a. Install efs mount helper in Amazon Linux. In the case of Amazon Linux this involves running the following command:
      `sudo yum install –y amazon-efs-utils`
      (This step was covered previously, but we're making a note of it here since there are multiple options for mounting the efs share.)
   b. Install nfs utils, if not using the efs mount helper. In Amazon Linux this would involve running:
      `sudo yum install –y nfs-utils`
      If using Amazon Linux, use the efs mount helper package.
   c. Create the shared directory mount point:
      `sudo mkdir -p /mnt/eft-share`
   d. Mount the share with a command like this:
      `sudo mount -t efs fs-321ec07a:/ /mnt/eft-share`
   e. If not using the efs mount helper, the mount command will look more like this:

      ```
      sudo mount -t nfs4 -o
      nfsvers=4.1,rsize=1048576,wsize1048576,hard,timeo=600,retra
      ns=2 fs-321ec07a.efs.us-east-1.amazonaws.com:/ /mnt/eft-
      share
      ```

   STOP: If you cannot mount the share, please ensure that you've attached the correct security group to both the Linux node and the EFS volume. The SG should have a rule that allows all traffic from itself.

   f. Optionally create an entry in /etc/fstab to persist the mount between reboots. Instructions for this may differ between using efs mount helper and nfs.
      Please refer to fstab documentation.

5. Next, CD into /etc/Samba and edit smb.conf using VI or similar editor. Add to the end of the file:

   ```
   [eft-share]
   path = /mnt/eft-share
   comment = EFS Volume
   valid users = eft-user
   read only = no
   create mask = 0660
   directory mask = 0770
   force group = eft-user
   force user = eft-user
   ```

Notice the mount point and username are the ones you defined in earlier steps.

6. Next, perform the following commands:

   ```
   $sudo chkconfig --level 345 smb on
   ```

```
$sudo service smb start
$sudo useradd -s /sbin/nologin eft-user –M
$sudo passwd eft-user
      Enter the password when prompted
$sudo chown -Rv eft-user:eft-user /mnt/eft-share
$sudo smbpasswd –a eft-user
```

Note: The above commands setup and start the smb service, creates a user using the username you supplied under smb.conf, assigns a password and then assigns that account to the smb service all in one fell swoop.

EFT and EC2 Setup

Before proceeding, you should have the following documented:

- The shared folder path, e.g. \\172.31.1.170\eft-share
- The user account name created earlier for Samba access
- The user account password created earlier for Samba access
- The AWS region, e.g. us-east-1

If so, we are now ready to perform EFT Setup.

EC2 Instance Prep

1. Navigate to the EC2 console and click **Launch Instance**.
2. **Look for a supported Windows instance (recommend the latest Windows 2016 or above).**
3. For **EC2 Instance Type**, choose a t* (medium or higher), if you expect to have odd traffic patterns (e.g., burstable traffic). If on the other hand, if you expect a consistent load, then choose an m* or c* instance with at least 4GB of RAM. EBS storage is OK, as storage will be streamed to an S3 bucket. Consult the Cost Calculator section for an idea on cost involved for each compute type.Cost Calculator
4. Under **VPC Setting**s, select or create a **VPC**. (Our example will use the default vpc.)
5. Under **VPC Settings**, choose a distinct subnet for the node and make note of it. Make sure each Windows instance you create lives in a different availability zone when selecting your subnet. For example, if the first node lives in us-east-1a, do not choose a subnet that lives in us-east-1a for your second instance.
6. For the Security Group, choose the security group you created earlier for EFT. If you have not yet created one, do so from here, adding the Inbound entries as mentioned in the earlier section.
7. Select or create a key pair that EFT will use to decrypt the instance password. You should select the keypair we created earlier.
8. Create the instance.

9. Repeat steps 1-4 above for the second and any additional nodes, with the following important sub-steps:
    a. Choose the *same* **VPC** as chosen for the other node(s). All nodes must be within the same VPC.
    b. Choose *different* **subnets** for each node. Each subnet maps to a specific, geographically disparate availability zone (AZs) within the same region. If you place two or more nodes in the same subnet, you are effectively creating a single point of failure. By placing nodes in different AZs you eliminate said single point of failure.
    c. Ensure you select the *exact same* security group for additional nodes. This group must have an inbound rule that has a Source equal to the security group ID, so that each node belonging to this group can talk to other nodes across AZs via their private IP.
10. Give the instances a few minutes to finish provisioning, then retrieve the Windows administrator password and connect to each instance via RDP. If it fails to connect, make sure you enabled RDP for the Security Group attached to this instance and ensure that the remote IP address is correct (e.g., your domain's external IP).
11. After you RDP in, check for Windows updates. The latest ami's may or may not be completely up-to-date, and now is the best time to check.
12. Optionally set windows updates to NOT auto-patch:
    d. Type Windows + R, type gpedit.msc to open Local Group Policy Editor, the location is Computer configuration, policies, Administrative templates, Windows Components, Windows Update, in the right pane click **Configure automatic updates > disable**.
13. Create a new user account (**Control Panel > User Accounts**) and password that is the same as the ones you created earlier for Samba access, i.e., eft-user: Eft1user!! (if needed, password complexity policy can be managed under **gpedit.msc> local computer > computer config > windows settings > security settings > account policies > password policy**). Note: it is important that the username and password for this user match the credentials for the file share.
14. In Windows, change the account type *from* Guest to **Administrator**.
15. Verify you can reach the shared path you created earlier. In Windows explorer, enter the shared path, e.g. \\172.31.1.170\eft-share, and enter the account credentials created earlier. Note: Accessing the share in this way does not make it available persistently. To make the share persistent between reboots, you can use the `net use` command as follows:

```
net use \\172.31.1.170\eft-share "Eft1user!!" /user:eft-user
/persistent:yes
```

STOP: If you cannot connect to the share then check your Samba and configuration and verify the drive is mounted and working and that there is a security rule allowing this Windows instance to interact with the Linux instance, as covered earlier.

1. Back on the first node, download the latest EFT Server installer from the Globalscape website: In your browser type: ftp://ftp.globalscape.com/bin/files/eftse/eftserver-ent.exe and hit Enter.
2. Right-click on the EFT installer and choose **Run as administrator**.
3. When prompted, choose **Active-active cluster** for the installation type.
4. **MSMQ** will be installed for load balancing of Timer and Folder Monitor Event Rules.
5. Choose **Yes**, if it is the first node in the cluster you are installing on.
6. For **Shared Settings Location**, specify the shared path you created earlier. For this example, it was \\172.31.1.170\eft-share.
7. Specify a username and password for the EFT administrator account when prompted.
8. On the Auditing and Reporting database configuration setup page, select **Configure Auditing and Reporting** if you plan on leveraging AWS's RDS or a separate SQL Server instance; otherwise choose **Skip**. Note that you can run the installer in **Modify** mode to add auditing in the future, if necessary.
    a. If you choose to configure auditing and reporting, choose **Use existing SQL Server.**
    b. Then choose **Create a new EFT Server ARM database**.
    c. On the database setup page, enter the **SRS endpoint** as the database server host address, and the username and password you created when SRS was configured (see earlier steps). Click **Test** to verify the connection.

        SRS endpoint: Reftdb.ccsmvm0w6g3x.us-east-1.rds.amazonaws.com

        Username: eftuser

        Password: Eft1user!!

        Database name: EFTDB

    d. Enter a database name (make one up), such as EFTDB, then repeat the username and password you used to connect to the DB.
9. Click **Next**, and **Next** again until the last wizard page appears.
10. On the last page, *clear the check box* for **Start the EFT Server** service account.
    Important Note: Do NOT attempt to run the EFT Service until the logon user account is configured in the next step.
11. Open the services applet (**services.msc**).
12. Locate the **EFT Server Enterprise** and double-click to open.
13. On the **Log on** tab, click **This account**, then click **Browse**.

14. Type in the account name you created back in the Samba share setup:

    Username: eft-user

15. Click **Check Names** to verify the account exists (if not, see earlier steps).
16. Click **OK** then **Apply** and enter the account password created earlier:

    Password: Eft1user!!

17. Click **OK** to the prompt regarding **Log on As a Service** right.
18. Click **Start** to start the EFT service.
19. Repeat the steps above on the second through Nth node, with the following differences:
    a. Choose **No** when asked if this is the first node in the cluster.
    b. Enter the same shared path as for node 1, so the node will find the configuration file.
20. After changing the logon account for the service, **Start** the service.

You have completed the manual setup process. Before configuring the load balancer, we will perform an initial setup of EFT's configuration to verify connectivity.

EFT Administration

EFT administration is covered in detail in the EFT help file http://help.globalscape.com/help/eft7-4/. Below is a quick reminder of the steps to setup a Server, Site, and User account, so that we can test each node's connectivity.

1. After completing manual setup of each EFT node, launch the EFT Administrator Interface on any given node, and choose **This computer** when prompted.
2. Enter the administrator username and password you created earlier.
3. Click **Start Trial** and start going through the initial setup wizard.
4. On the **SMTP** tab, optionally enter your Amazon Simple Email Service (SES) connection information and credentials, or those of a mail relay you have full access to, or skip that section (leave defaults).
5. After completing the initial setup wizard, run the **Site setup wizard**.
6. Choose **GSB auth** or optionally configured AD services (see ADS towards the end of this document).
7. Enable **HTTPs** over port **443** and, optionally, **SFTP** over port **22**
8. Configure **SSL and SFTP certificates** when prompted.
9. Complete the Site setup wizard then launch the **User setup wizard** (unless using AD).
10. Create a test user following the wizard prompts (unless using AD).

11. Now open the **Windows Firewall** utility (wf.msc) and add an **inbound rule** for ports 443 and 22, or create an inbound rule for all ports to the EFT service executable (C:\Program Files (x86)\Globalscape\EFT Server Enterprise\cftpstes.exe)
12. **Verify** setup by pasting in the public DNS name of the first node (find in EC2 console) into your local browser, preceded by **https://.** Login using the username and the password you specified for the user account. Proceed to the Troubleshooting section if unable to connect.

    Note: You will need to modify the EFT Security Group to allow access over 443 to your local IP to verify setup. Otherwise, wait until the LB is configured and then verify setup.

After completing the setup process and performing basic administration, we are ready to setup the load balancer.

AWS ELB Setup

Amazon's Elastic Load Balancer (ELB) service is used to direct traffic to each node in the cluster in round-robin fashion. Setting up the LB requires a couple of preparatory steps, covered next.

Certificates

If you are planning to use HTTPS, the ELB service will require a copy of the certificate pair used by EFT to decrypt HTTPS traffic. While it is possible to perform TCP load balancing in AWS of HTTPS traffic, the LB won't allow configuration of node affinity (session stickiness) options unless it can decrypt the https traffic. Without node affinity, EFT's normal redirection upon login will fail due to the LB's intrinsic round-robin behavior.

Thankfully, EFT's setup wizard will generate a self-signed cert for us. You can create a cert populated with sample values that are suitable for evaluation or dev environments and for verifying that LB works; however, for staging or production environments, you should create your own public and private key pair, rather than use the sample ones. A key pair including Certificate Signing Request (CSR) can be created from within EFT (search EFT's help topic for creating SSL certificates) or via Amazon's certificate creation tool, or third-party tools such as OpenSSL. Once you create your own key pair, have it signed by a CA, and then import the public and private key into EFT, as described in EFT's help file.

PEM Formatting for Aws Certificate Manager

AWS only accepts *pem*-encoded private keys. While EFT can create PEM certificates, you may have to convert your keyfile into an RSA private key to import it into ACM. Once converted, we will import the private key and public certificate into AWS's certificate manager tool. The following instructions assume that you created an SSL cert using EFT's Site Setup Wizard. It also assumes you created a cert and private key in PEM format.

When generating pem ssl keys/certificates with EFT, follow these steps to obtain an rsa formatted private key for use with AWS ACM:

1. Launch Putty and connect to the Linux instance you created earlier, if disconnected.
2. Once connected, navigate the shared directory and list its contents

   ```
   $ cd /mnt/eft-share
   $ ls
   ```

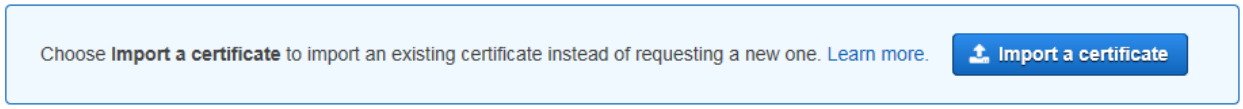
3. You should see a file named "MySite Certificate.key" or similar (with .key extension)
4. Run the following command to output the private key to a pem file.
5. ```
   $ sudo openssl rsa –in "MySite Certificate.key" –out "MySite Certificate.key.rsa"
   ```

   Note: The password will be the Instance ID for the first node in the cluster. You can obtain this information from the EC2 > Instances console, under the Description tab for this node.

   STOP: If you encounter an error you will not be able to import the cert into AWS.
6. Perform another "ls" to verify the presence of the .key.rsa file
7. Open the .key.rsa and .crt file into a text editor so you can see and later copy the contents.

Certificate Import

1. In AWS, navigate to **Services** > **Certificate Manager**.
2. Click **Get Started** if this is the first time managing certs.
3. Click on **Import Certificate** on the main page.



4. Carefully copy the contents from the .crt file into the Certificate body block
5. Repeat the procedure, copying from the .key.rsa file into the Certificate private key block

Note: A single extra space can cause the import to fail. Copy from the first dash in the BEGIN sequence to the very last dash (inclusive) in the END sequence.

STOP: If you encounter an error you will not be verify that the certs are in the right format.

6. Click on **Review and import** to complete the process.

ELB Setup

1. From within the Amazon AWS console, navigate to EC2 > **Load Balancers**.
2. Click **Create Load Balancer**.
3. When prompted, click **Classic**.
   ```

4. Give the LB a **Name** and designate the **Scheme** as internal-facing, ipv4.
5. Under **Listeners**, choose the listeners that correspond to the protocols you will be exposing from within EFT, such as **HTTPS** over 443 and **TCP** over 22. Note that if you add more, you will also need to update EFT's connection settings, Windows Firewall, and the AWS Security Group.
6. Choose the **VPC** where the cluster resides.
7. Click on **Enable advanced VPC configuration**.
8. Add the Availability Zones (subnets) within the region where each node resides. There should be at least two unless you accidently put both nodes in the same subnet earlier. (Go back and fix it.)
9. On the next page, select the same **Security Group** you created earlier for the ELB.
10. On the **Configure Security Page**, choose the certificate in the drop-down list that matches the one you imported earlier. You may optionally upload a new certificate here; however, sometimes the LB creation will fail if the cert takes a while to propagate.
11. The **Predefined Security Policy** should be fine, or make one of your own (so long as ciphers, protocols, and options match what is available and enabled in EFT).
12. In the **Backend Certificate** section, click **Enable backend authentication**. You'll want to add a copy of your certificate here and give it a name. The load balancer communicates with an instance only if its public key matches this key.
13. On the **Health Check** page, change the **Ping Protocol** to HTTPS, port to 443, and **Ping Path** to '/EFTClient/Account/Login.htm'
14. On the **Add Instances** page, select the EFT nodes you created earlier.
15. Finish creating the LB.
16. Back on the LB dashboard, select the newly created LB and ensure the **Description** tab is highlighted.
17. On the **Description** tab, under the Port Configuration section, click **Edit stickiness**.
18. Change the default value (Disable) to **Enable LB generated cookie stickiness**.
19. Leave the session expiration blank (or specify a value if desired) and save your changes.
20. Back on the **Description** tab, copy the **public DNS** name for the LB to your clipboard.
21. Paste the LB public DNS name into your local browser, preceded by https:// to test out the setup, or test using an SFTP client like CuteFTP®.

STOP: If you cannot connect then either the LB is not configured correctly or an earlier step was missed. Make sure you can connect to the instance (node) public IP directly, and if that works, then the LB is misconfigured. If direct connections to the node fail then consult the troubleshooting section.

Note: If the connection worked then you are done. The next section deals with setting up RDS and then alternate method of deploying EFT using the manual, rather than scripted approach.

Advanced Configurations

DMZ Gateway

Introduction

Deploying a 1:1 ratio of DMZ Gateway to EFT nodes provides maximum security. The DMZ Gateway is a sophisticated type of reverse proxy that provides the following benefits:

- It prevents data from every being stored in a public or demilitarized zone (even temporarily), to comply with security regulations such as PCI DSS, which do not allow protected data from residing in the DMZ.
- It negates the need for directory services (for user auth) in the DMZ of your network or public zone. This is important for admins that would rather not expose their Active Directory services to the world.
- It removes the need for ANY inbound connections from the DMZ to the trusted zone, with the exception of locked down (by IP) access to RDP or SSH for node administration. How is this possible? This is possible because connections are initiated from EFT (outbound) to the DMZ Gateway, where it maintains an ongoing "Peer Notification Channel" (PNC). When external (client) connections are made to the DMZ Gateway, the DMZ Gateway notifies EFT over the PNC. EFT subsequently opens another *outbound* connection to the DMZ Gateway, which then "glues together" the external connection to the new one made by EFT. As such, even if the DMZ Gateway server were to be compromised, no *inbound* connections could be made to the trusted zone, where directory services, file storage*, etc. are maintained.

For more information regarding the benefits of the DMZ Gateway and the differences between it and "normal" reverse proxies, please refer to the DMZ Gateway help guide: http://help.globalscape.com/help/dmz3/introductiontodmzgateway.htm

*S3 storage is technically externally addressable. This is addressed under the Security Consideration topic.

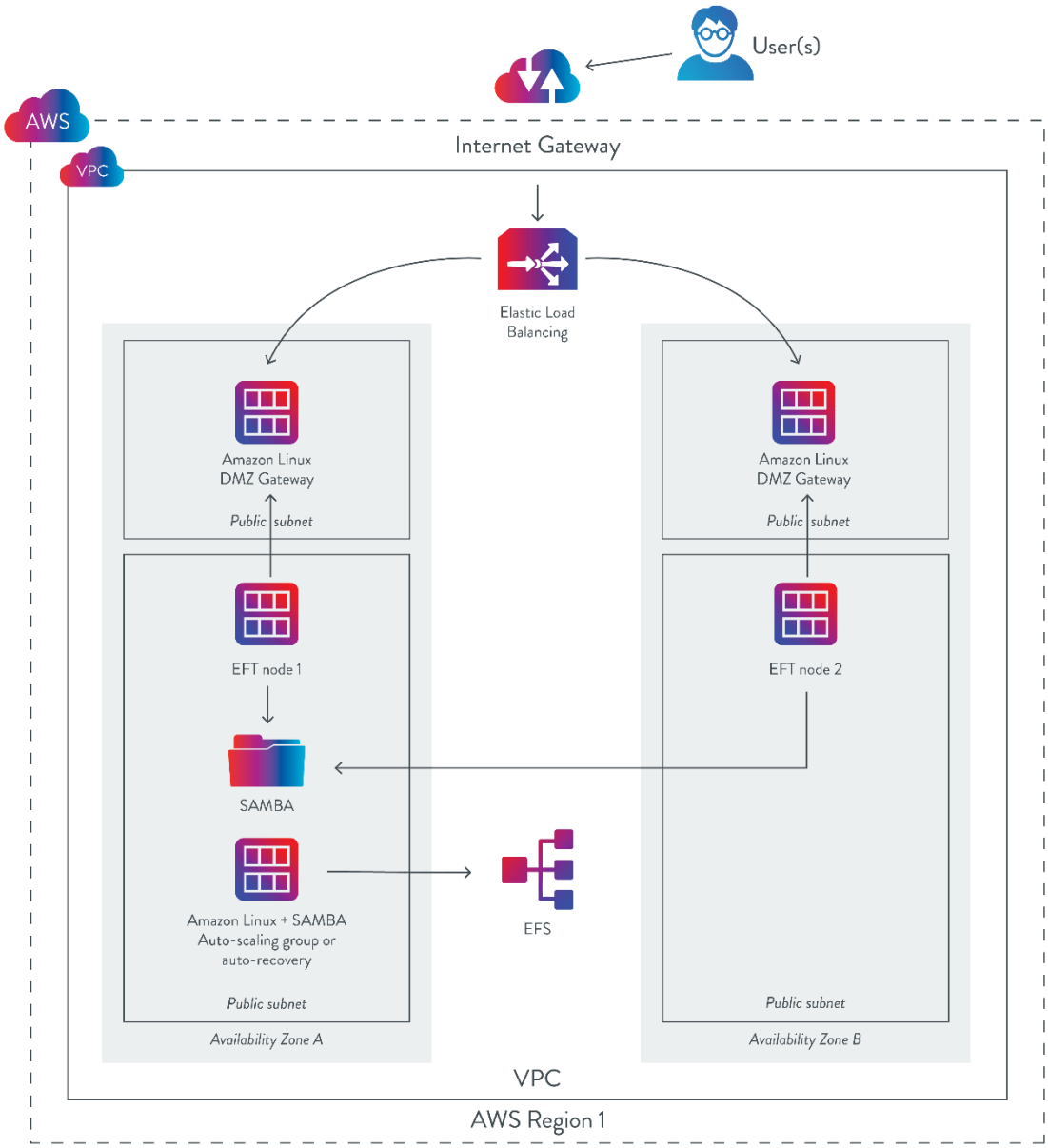
Architecture

The following diagram expands upon the diagram shown earlier in this document. Note that for maximum performance and value (lowest cost), Amazon Linux AMIs are used for the DMZ Gateway nodes. This will require headless administration of the DMZ Gateway, which is covered in its help file here: http://help.globalscape.com/help/dmz3/dmz_gateway_headless_administration.htm.

A few key points of observation regarding this architecture:

- Security Groups will need to be modified to lock down direct access to EFT and instead open access only to DMZ Gateway.
- An additional subnet is added in each of the two AZs

- The ELB needs to be modified so that it points to each DMZ Gateway node, rather than to each EFT node.
- Consideration for inbound administration ports and listener ports needs to be made, as SSH typically listens on 22, which happens to be the SFTP inbound port.
- Costs will increase due to the additional Linux nodes and DMZ Gateway licensing fees.
- The subnets are still public, so that the administrator can access/manage EFT and the OS it runs on. This is discussed in more detail in the Security Considerations topic.



DMZ Gateway® Setup

Assuming you are familiar with and has chosen to deploy DMZ Gateway alongside EFT, the following section will detail how to do so in AWS in an HA clustered environment according to the diagram shown above.

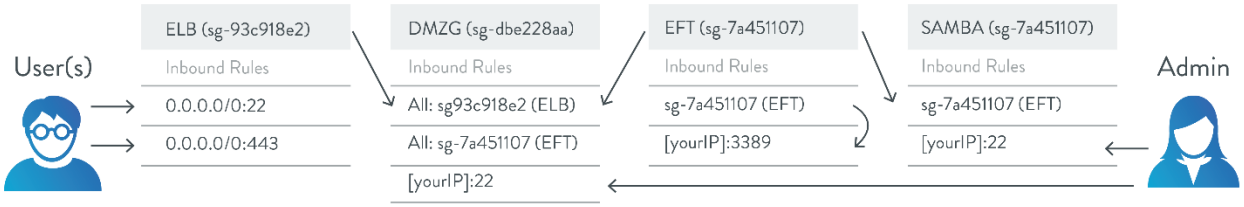
Subnets

It is acceptable to create the DMZ Gateway Linux AMI instances in the same subnet corresponding to its peer EFT. However, for this exercise we will create these instances in a separate subnet, as this will make it easier if you decide to make the EFT subnet private and leverage a VPN or jumpbox to manage the EFT node in the future. This is discussed in more detail in the Security Considerations topic. If you prefer to use your existing, default subnets, you may do so and skip this section.

1. In the AWS console, navigate to the **VPC** dashboard, then to **Subnets**.
2. Under subnets, you should see a list of default subnets, with one created for each availability zone (AZ) within the current region. Click **Create Subnet**.
3. Give the subnet a name, such as **DMZG AZ A**, and specify the *same* VPC as you've been using.
4. Under AZ, specify one of the AZs that has an EFT node present and document it.
5. Specify an **IPv4 CIDR block** that doesn't overlap with an existing one in the same AZ. For example, if our VPC IPv4 CIDR is 172.31.0.0/16 and 172.31.64.0/20 is the last CIDR block specified of the 5 default subnets provided, then we could add a CIDR block at 172.31.80.0/22 to create an addressable space of ~1,000 IPs for that subnet.
6. Create the subnet, then select the subnet from the list and click **Subnet actions**.
7. Select **Modify auto-assign IP settings** and select the **Enable auto-assign public IPv4 address** check box. If you don't select the check box, you won't be able to administer the node.
8. Repeat steps 1-7 above, and create a subnet for DMZG AG B, with an appropriate subnet CIDR block, such as 172.31.84.0/22, also enabling auto-assign IPs.

Security Groups

Introducing the DMZ Gateway changes the security posture. EFT no longer has to be open to the world (with the exception of the RDP port, which is locked down to your IP), as EFT will establish an outbound connection to the DMZ Gateway. Here is what the groups will look like once you are done:



Load Balancer Security Group

1. The ELBSecGroup will remain the same as it was:
    a. HTTPS, TCP, 443, 0.0.0.0/0
    b. SSH, TCP, 22, 0.0.0.0/0

DMZ Gateway Security Group

1. Add a new Security Group named "**DMZSecGroup**" or similar.
2. Create the following rules:
    a. SSH, TCP, 22, [YOUR_IP] (or 0.0.0.0/0, but much less secure!)
    b. All Traffic, All, All, sg-[ELBSecGroupID]
    c. All Traffic, All, All, sg-[EFTSecGroupID]
3. Click "**Save**"

Windows (EFT) Nodes Security Group

1. Edit the "**EFTSecGroup**" group as follows:
    a. RDP, TCP, 3389, [YOUR_IP] (or 0.0.0.0/0 much less secure!)
    b. All Traffic, All, All, sg-[EFTSecGroupID] (e.g. itself!)
2. Click "**Save**"

Linux Node Security Group

1. The LinuxSecGroup will remain the same as it was:
    a. SSH, TCP, 22, [YOUR_IP] (or 0.0.0.0/0, but much less secure!)
    b. All Traffic, All, All, sg-[EFTSecGroupID]

These groups allow users to connect to the ELB, which in turn communicate with the DMZ Gateway. The DMZ Gateway also accepts connections from EFT and the Linux node (Samba share). Inbound connections to your cluster will not work until after you've configured the DMZ Gateway nodes.

Note: If your EFT HA cluster is actively handling traffic, then you may want to create the DMZ Gateway security group, but hold off on modifications to the other groups until after you've setup the DMZ Gateway nodes and have modified the ELB to route traffic to the DMZ Gateway nodes. Otherwise, inbound traffic to EFT via the ELB will fail once you save the Security Group changes shown above.

Create Linux AMIS

1. **In the AWS marketplace, locate a Linux AMI, such as ami-c58c1dd3 (for US-East-NV)** Amazon Linux AMI (HVM / 64-bit).
2. Launch one Linux AMI instance for *each* DMZ Gateway node:
    a. Choose an appropriately sized t*, m*, or c* instance. This instance does not need much disk space or memory, as it will simply broker connections from the ELB through to EFT. For this exercise, we choose an M4.Large.
    b. Ensure that each Linux instance is associated with the corresponding subnet you created above (or the existing subnet for each EFT if re-using), such that there is a ratio of 1:1 EFT to DMZ Gateway nodes in in the same AZ. Do not accidently crisscross them as this would unnecessarily eat up inter-AZ bandwidth costs.
    c. For the Security Group, choose the one you created in the earlier step (DMZSecGroup)
    d. Finish launching the instance.
3. Record the private IP of the Linux instance, as you will need it later. For example:

    File Share IP: "172.31.1.170"

Install DMZ Gateway

Repeat the following steps on both Linux nodes:

1. SSH into Linux using Putty or similar. Guide here: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html
2. Down the latest DMZ Gateway using wget or curl, for example:

```
$sudo curl ftp://ftp.globalscape.com/bin/files/dmz/dmz-gateway-linux-x86-64.tgz
-o /dmzg.tgz
```

3. Next, unzip and untar the setup files, then run the installer:

```
$sudo gunzip /dmzg.tgz
$sudo tar xvf /dmzg.tar
$sudo ./Install.sh
```

4. Type Yes to accept EULA, ENTER for all the defaults, Yes to setup service.
5. Choose Yes when prompted to register the DMZ Gateway service.
6. Types Yes to run the DMZ Gateway so that the gwconfig.xml gets created
7. *Optionally* stop the DMZ Gateway service (`$sudo /opt/dmzgateway/bin/dmzgatewayd stop`) if you want to edit the gwconfig file. (Refer to the DMZ Gateway online help for headless administration: http://help.globalscape.com/help/dmz3/dmzgatewayserverconfigurationfilegwconfig.xmlreference.htm)

8. Make sure you repeat the process on both nodes, thus standing up a DMZ Gateway service for each EFT node.

Configure EFT

The final step is to configure EFT so that it connects to the DMZ Gateway server to receive inbound connections.

1. RDP into one of the EFT nodes, making node of its particular subnet and AZ.
2. Launch the EFT administration interface.
3. In the EFT administration interface, expand your Site in the tree, left pane.
4. Click the **DMZ Gateway** node in the tree.
5. On the **DMZ Gateway** tab in right pane, provide the private IP address of the DMZ Gateway server that corresponds to that same AZ, whether it is in the same or a different subnet. Do not accidently point an EFT in one AZ to an EFT in a different AZ, as this will incur inter-AZ bandwidth charges.
6. Enable the desired protocols and ports. In this example we used HTTPS (443) and SFPT (22) throughout; however, because the Linux nodes use port 22 for administration (SSH), change the SFTP port to 2222 (or similar, but not 22). Later, we will modify the ELB to route incoming client connects on port 22 to port 2222 on the DMZ Gateway server. You do NOT need to modify EFT's SFTP listening port under Site > Connection settings, ONLY under the DMZ Gateway settings.

IMPORTANT: Specify port 2222 under the SFTP port under DMZ Gateway connection settings.

STOP: If you do not see a green "connected" icon next to the DMZ Gateway server, check your Security Group settings and verify that you followed the DMZ Gateway setup instructions so that EFT can connect to the DMZ Gateway node without errors.

Note: If you see the green "connected" icon, then a connection has been established to the DMZ Gateway and you are ready to modify the ELB to start routing traffic to the Gateway nodes.

Note: You may optionally remove the two firewall rules you created on the Windows system as EFT will be brokering connections through the DMZ Gateway, rather than directly.

Configure ELB

The final step is to alter the ELB configuration (see ELB setup section) so that the ELB routes connections to the DMZ Gateway nodes, rather than to the EFT nodes.

1. From within the Amazon AWS console, navigate to **EC2 Dashboard**.
2. Navigate to the Load Balancing subsection and select the ELB you created earlier.
3. Under the **Instances** tab, click **Edit Instances**, clear the selection of the two EFT instances, and then select the two DMZ Gateway instances.

4. Under the **Listeners** tab, click **Edit**, and configure the port mapping as follows:

LB protocol	LB port	Instance Protocol	Instance Port
TCP	22	TCP	2222
HTTPS	443	HTTPS	443

This simply maps incoming SFTP connections over port 22 to the DMZ Gateway's port 2222.

Note: You may need to modify cipher settings if HTTPS connections fail, to ensure the client and server can negotiate an agreed upon cipher suite.

5. Optionally edit the Health Check to point to 2222 or continue to use 443 and the ping path configured earlier.
6. To test the setup, paste the LB public DNS name into your local browser, preceded by https://, or test using an SFTP client like CuteFTP or similar.

STOP: If you cannot connect then either the LB is not configured correctly or an earlier step was missed. Consult the troubleshooting section for more help.

Note: If the connection worked then you are done. The next sections cover optional services including ADS and SES.

ADS

Intro

EFT offers several choices for authenticating users, ranging from a powerful, built in authentication manager, to RSA token-based authentication, to Active Directory authentication. For customers that wish to leverage AD, Amazon lets you choose between standing up your own AD domain controller in an EC2 instance, or leveraging Amazon's Active Directory as a service. There are plenty of resources online that discuss the pros and cons of each; however, in this instance we will proceed with the as-a-service option, as attaining high availability is the primary purpose of this HA cluster.

Setup

1. From within the Amazon AWS console, navigate to **Directory Services**.
2. Under **Microsoft AD**, click **Set up Directory**.
3. Follow the onscreen instructions (setting up and configuring AD is outside the scope of this document, but is covered extensively in AWS documentation).
4. RDP into one of the EFT nodes.

5. Launch EFT's administration interface and authenticate as admin.
6. In EFT, create a Site that uses AD and enter the AD lookup and authorization credentials to connect and retrieve a list of users. For more help with AD setup in general, refer to EFT's online help documentation. http://help.globalscape.com/help/eft7-4/Creating_a_Site.htm#AD

SES Configuration

Intro

Amazon's Simple Email Service allows you to relay emails off AWS originating from EFT. To setup SES, navigate to SES in the AWS console and follow the instructions. Back in EFT, on the **Server** node, click the **SMTP** tab, provide the mail host address and login credentials, enable SSL, and then send a test email to verify proper configuration.

Security Considerations

This guide provides a relatively high-level setup procedure to get an HA cluster up and running, while providing basic security using AWS Security Groups. It is up to the reader to understand and implement proper security controls such that best practices, internal policies, and regulatory standards are met, and to ensure that the attack surface is reduced as much as possible.

Security considerations should include but are not limited to:

1. Generate your own set of certificates and have them signed by a CA, then pair them with EFT and the ELB. The certificates generated by EFT are self-signed and thus are not considered secure.
2. Set up and use the DMZ Gateway alongside EFT (covered in the Advanced Configuration section).
3. Create jump boxes for administration of nodes, such that the nodes themselves can be isolated to a private subnet, with the exception of the DMZ Gateway node, which must be reachable by the ELB (or EFT, if deploying without the DMZ Gateway).
4. Alternatively, consider leveraging either an AWS hardware VPN or Software VPN to administer the individual nodes: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpn-connections.html
5. Consider using VPC Endpoints (http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html) to isolate your S3 objects stores such that they can be reached ONLY from the private IPs of the Linux node that is running Samba.
6. Setup procedures for back-up and disaster recovery.

Troubleshooting

The following table describes possible problems and suggested resolutions:

Problem	Resolution

Unable to Connect to EFT instance directly using web browser	• Verify EFT is running
	• Verify EFT protocols and ports defined properly in Connection Settings
	• Verify EFT services (cftpstes.exe) using "netstat -abon -p TCP" command in Windows
	• Verify inbound FW rules are setup in Windows
	• Verify AWS Security Group allows access
	• Verify the public IP address for the node
	• Verify HTTPS was used, not HTTP
	• Verify correct username and password were used
	  o Either defined in manual deployment
	  o Or "User" and "[Instance-id]" if auto-deployed

Unable to connect to EFT instance via Load Balancer IP	• Verify ELB is running. • Verify ELB shows that instances are up and running. • Verify that EFT connectivity is working for direct connections (see above).
Unable to access shared path from Windows	• Verify that the Linux node's AWS Security Group has an inbound rule that permits the EFT node's Security Group to access the Linux node. • Verify that file share is mounted (run $sudo mount command in Linux). • Verify you are attempting to connect to the mount point excluding the mount root • Verify that the Samba service is running. • Verify that you created the appropriate permission entries in the Samba conf file • Verify that you are using the same account you assigned to the Samba share in Windows, and that the account is assigned to EFT as the **Log on as** account. • Verify that the IAM role you created earlier was given S3 full privileges.
Unable to run manual setup without errors	• Verify each step was followed carefully • Verify the latest version of EFT was installed (versions prior to 7.3.6 do not support HA clustering in AWS). • Verify that the sequence for setting up SQS and deploying the AwsConfig.json file was followed.

Problem	Resolution
EFT setup OK but is not clustered (can't see nodes in node list).	• Verify that the json file was copied and pasted without breaking the json format (use third-party tools to confirm format when in doubt). • Verify the json file key:value pairs were properly updated with the correct values for the SNS ARN, queue prefix, IAM access keys, and so forth. • Verify that the IAM role you created earlier was given SQS and SNS full privileges. • Verify that the SNS topic was created per the instructions. • Verify under the SNS topic that a corresponding message queue was subscribed to the topic and that RAW mode was enabled for each queue. • Verify that the SQS queue names followed the appropriate format of queue name prefix _ node name, such as EFTHA_NODENETBIOSNAME.
EFT is clustered but event rules are not load balancing.	• Verify that the etc/hosts file on each node was updated. • Verify that the correct private IP and NetBIOS names were entered into the host file so that each node can find all other nodes via its NetBIOS name (which is required for Event Rule load balancing). • Verify that the Security Group(s) to which each EFT node is attached has an inbound rule that permits the Security Group for other nodes to access that Security Group. Even if both nodes belong to the same Security Group, make sure the Security Group can access itself via an inbound rule entry. Remember that each node is in a difference AZ and thus require a rule so they can see each other.
For all other errors:	It is up to you to troubleshoot other problems related to deploying EFT in the AWS environment, as Globalscape's support services are generally limited to troubleshooting EFT application configuration issues. Many of the services outlined in this document are provided by third parties and are not the responsibility of Globalscape. Please read the following *Disclaimer* for more.