Primer

Primer Delta Lite

Administrator Guide

Primer Federal, Inc.

27 February, 2023

Version: 1.0.2

Table of Contents

Introduction

What is Primer Delta Lite?

Primer Delta Lite transforms unstructured search results into structured data for more efficient and effective exploration. Primer Delta Lite offers several capabilities:

- Extract and aggregate important data about people, organizations, etc.
- Cluster similar documents into broad topics and time-bound events
- Summarize documents, topics, events, and the entire result set
- Organize the data for easy exploration

Primer Delta Lite is a version of Primer's Delta product, packaged for easy, push button deployment in Amazon Web Services (AWS). It integrates with AWS S3 and SQS for easy import of your data into Delta. Primer Delta Lite includes an integrated Keycloak identity and access management (IAM) solution, which enables management of user accounts inside the solution and/or integration with your enterprise IAM solution.

How does Primer Delta Lite work?

Primer Delta Lite reads documents from its internal document store, extracts structured data, clusters documents into events and topics, and summarizes sets of documents. The first step is to import your data into the internal document store. Users of Delta can then query the internal document store using a boolean search form and then run an analysis on the resulting set of documents. Users can browse the analysis once processing is complete.

It takes Delta a few seconds to read each document the first time. Documents that have been read before will be cached and will process much faster. In order to keep analysis run times reasonable, there is a cap in Primer Delta Lite on the maximum number of documents that are processed in any analysis. See CloudFormation Parameters Reference for information on maximum result set sizes. Primer Delta Lite is configured to support running on analysis at a time.If users submit multiple analyses at the same time, they will be placed in a queue and each will wait its turn to be processed.

What is covered in this guide?

This guide is for administrators of Primer Delta Lite. It covers the administrative lifecycle, from ordering the application in the AWS Marketplace to running your first instance of Primer Delta Lite through typical application management tasks.

For information on how to use Primer Delta Lite, please see these resources:

- Training Videos: Link TBD
- FAQ and contextual help available within Primer Delta Lite

Getting Started

Ordering Primer Delta Lite

Primer Delta Lite is an AWS Marketplace offering by Primer AI. From the Primer Delta Lite product page in the Marketplace, subscribe by selecting the **Continue to Subscribe** button. On the next page, review and select **Accept Terms**. Once the subscription process is complete, you will be given a link to launch the product. You may also later access your subscription from the AWS Console, by navigating to **AWS Marketplace Subscriptions**.

Running Primer Delta Lite

Primer Delta Lite is an AMI based product delivered via CloudFormation. Running Primer Delta Lite means launching a CloudFormation stack that will create a Virtual Private Cloud (VPC), run an instance of the AMI, create an S3 bucket and SQS queues for the application, and set up all the roles, policy, and networking to connect everything.

Pre-requisites

Before launching the CloudFormation stack, ensure you have sufficient permissions. Either:

1. Have a role assigned to your AWS account with sufficient permissions to perform all the actions in the CloudFormation template.

2. Have a service role defined for CloudFormation that has sufficient permissions.

As an example, a user role that is assigned the AWS managed *AdministratorAccess* policy would be able to launch the CloudFormation stack.

Alternatively, one may create a new role with the AWS CloudFormation service as the role's trusted entity. That role could then be assigned the *AdministratorAccess* policy or assigned a new policy that grants least privilege permissions. See CloudFormation Stack Least Privilege Policy in the Appendix of this guide for that example policy.

You should also have a keypair registered in AWS in the same region where you will launch the stack. That key can be used to SSH to the running EC2 instance to perform any maintenance tasks on the server.

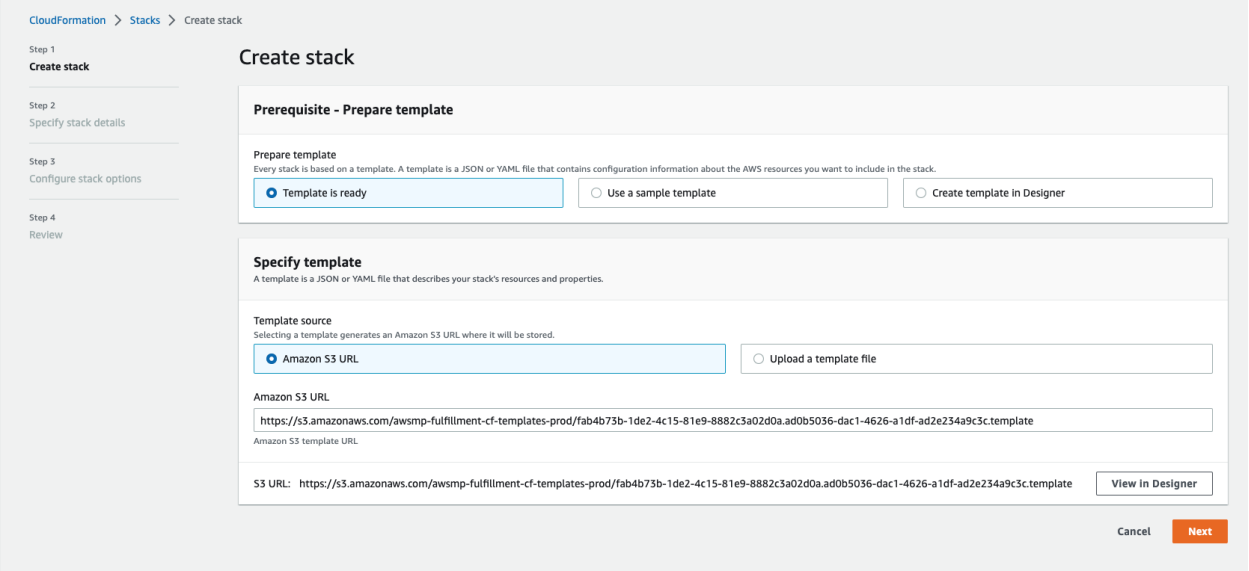
Creating a Stack

1. In the AWS Management Console, navigate to the **AWS Marketplace Subscriptions** service

2. Locate your Primer Delta Lite subscript and click **Launch CloudFormation Stack**

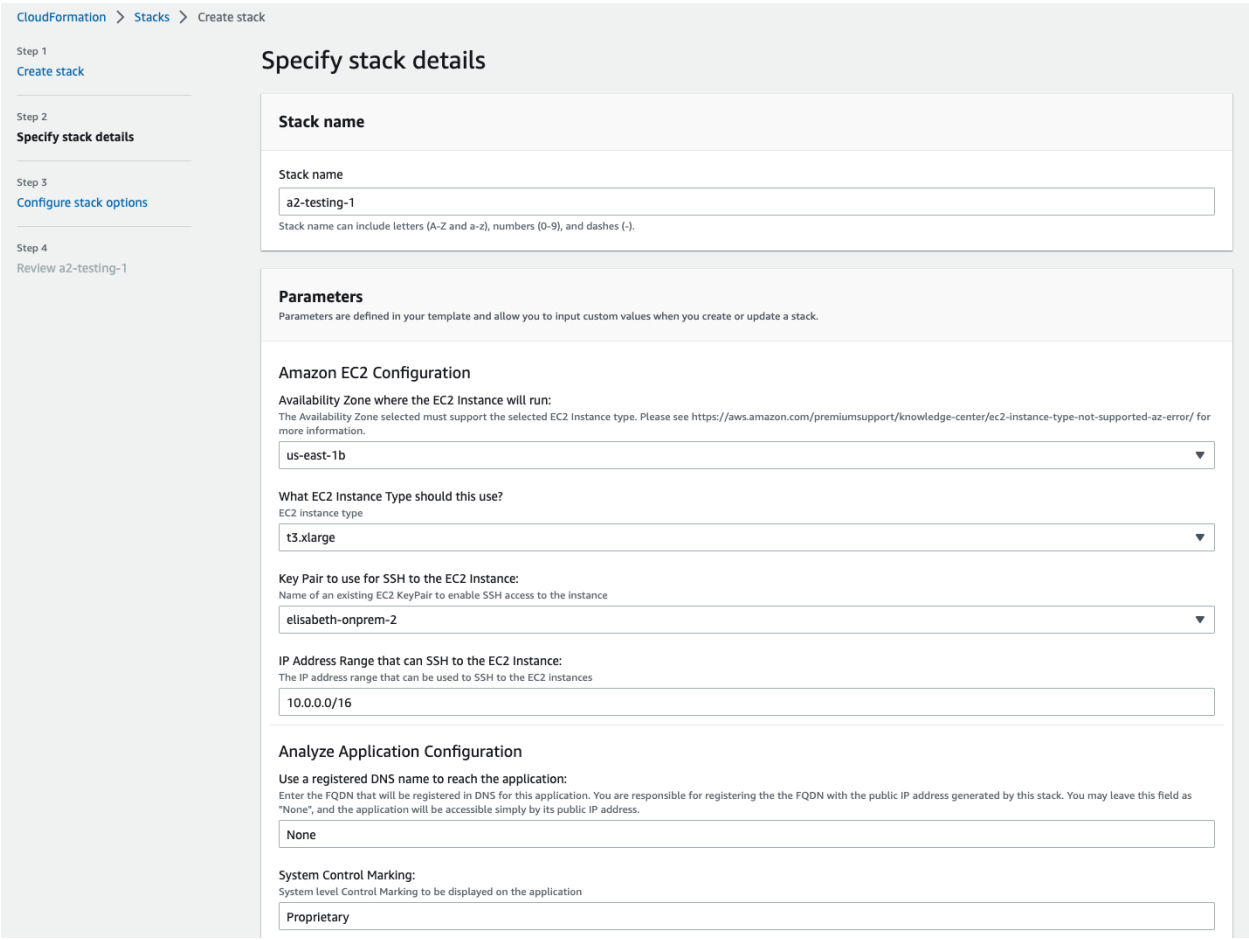
3. FIll in an submit any Marketplace forms for the product and select **Launch CloudFormation**

4. This will take you to the **Create Stack** screen in the CloudFormation console. The form will be pre-filled with the correct template URL. Click Next

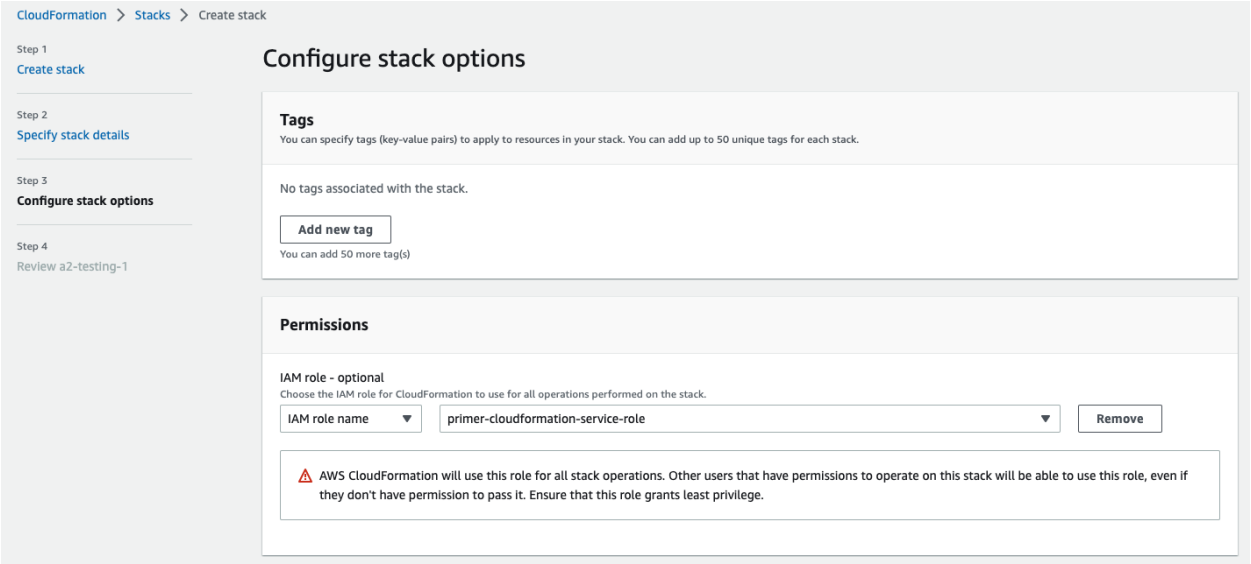


5. Enter a unique name for your stack. A stack name can contain only alphanumeric characters (case-sensitive) and hyphens. It must start with an alphabetic character. Use a name no longer than 40 characters.

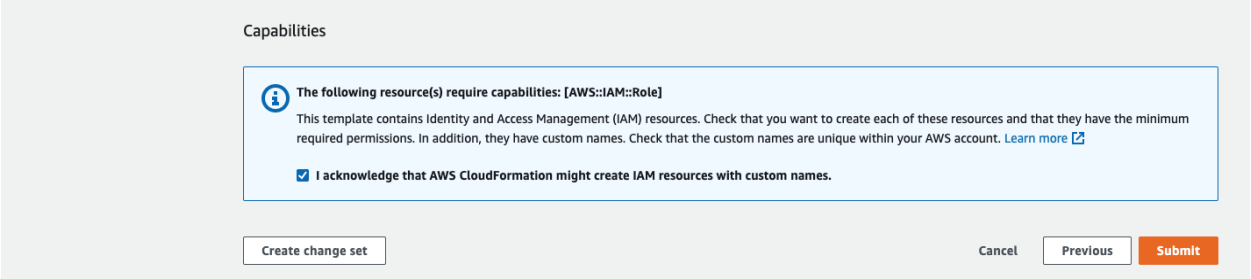
6. Fill in the **Parameters** form as needed. See CloudFormation Parameters Reference in the Appendix of this guide for more information on each of these parameters.



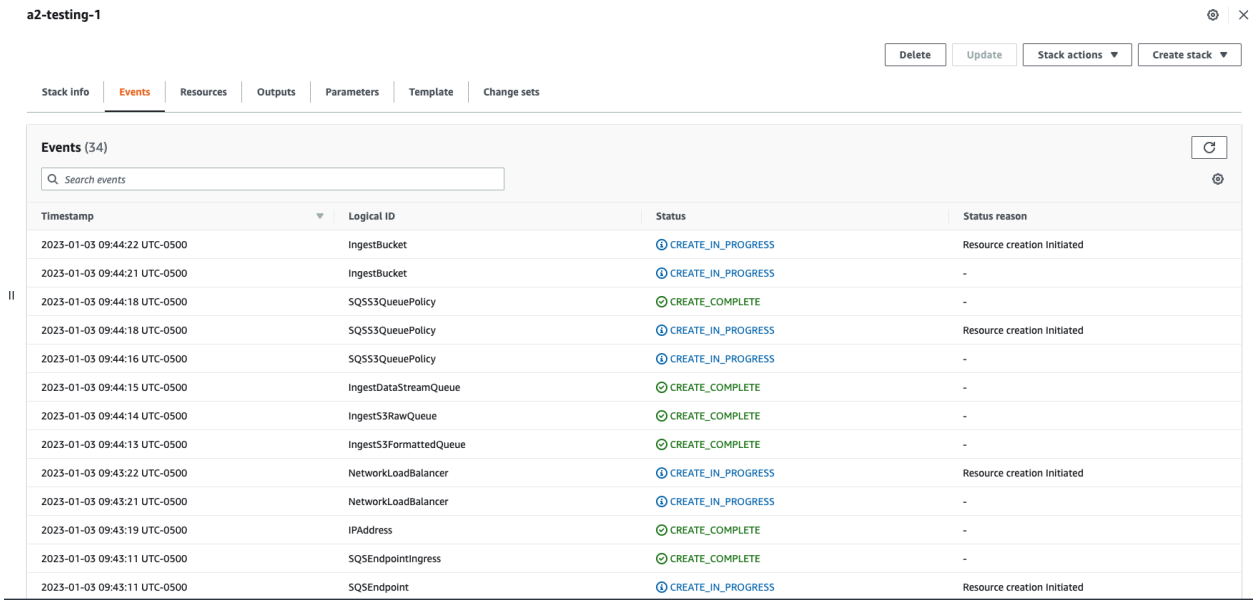
7. Click **Next**. If your AWS user account does not have sufficient permissions to create all resources needed by the stack, select a CloudFormation service role in the **IAM role** form that does have sufficient permissions.



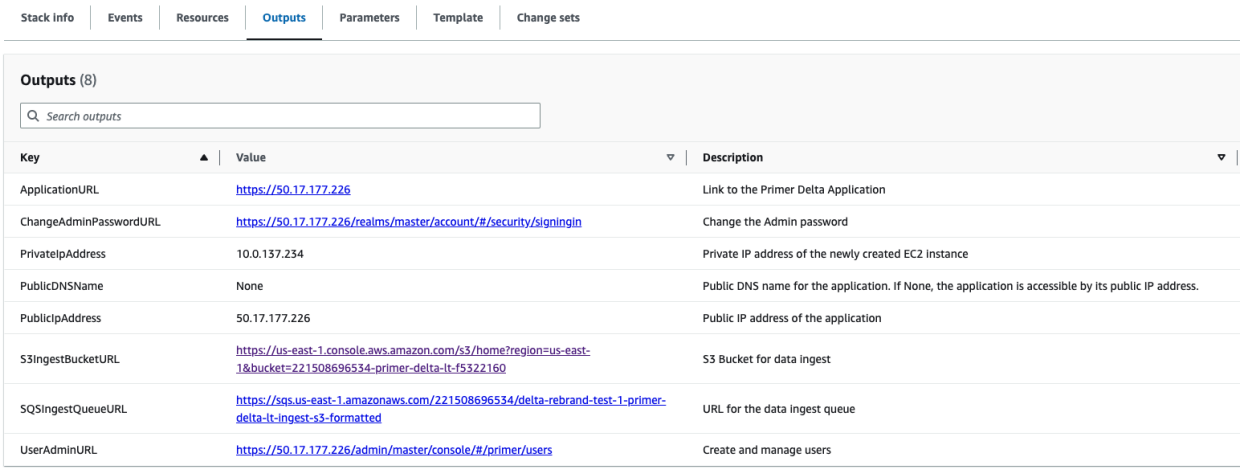
8. Scroll down and click **Next** again, scroll down and check the IAM Resources acknowledgement, and finally click **Submit**.



9. The console will display a progress screen for the stack installation. You may update the view with the refresh button on the upper right. The provisioning process takes about 15 minutes to complete. The stack creation will complete after 5-10 minutes. During the remaining time the EC2 instance will boot up and the application will perform its one-time configuration process.



10. After provisioning is complete, consult the **Outputs** tab for your stack. This provides some helpful links to use and administer your new instance. The **ApplicationURL** link will be functional once the application has successfully installed and configured itself.

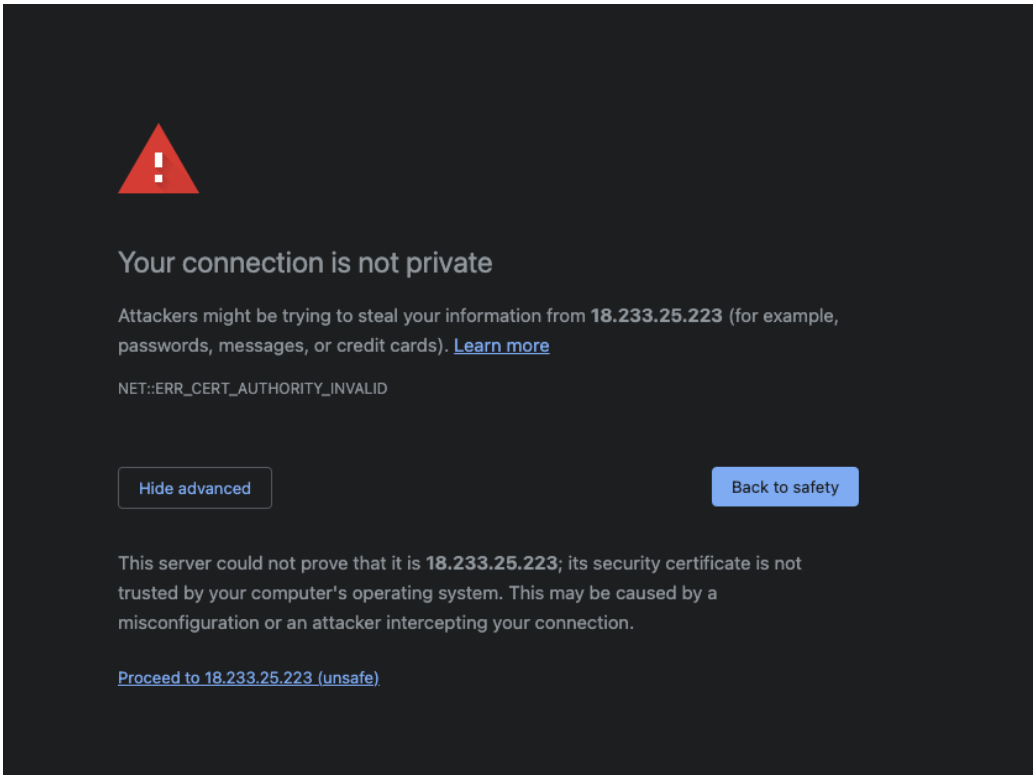


11. [OPTIONAL] If you entered a DNS name for the application during stack creation, register or update the name in your DNS system using the **PublicIpAddress** created by the stack. Further, obtain and install a valid PKI service certificate for that domain name. See Installing a Server Certificate in this guide for more information.

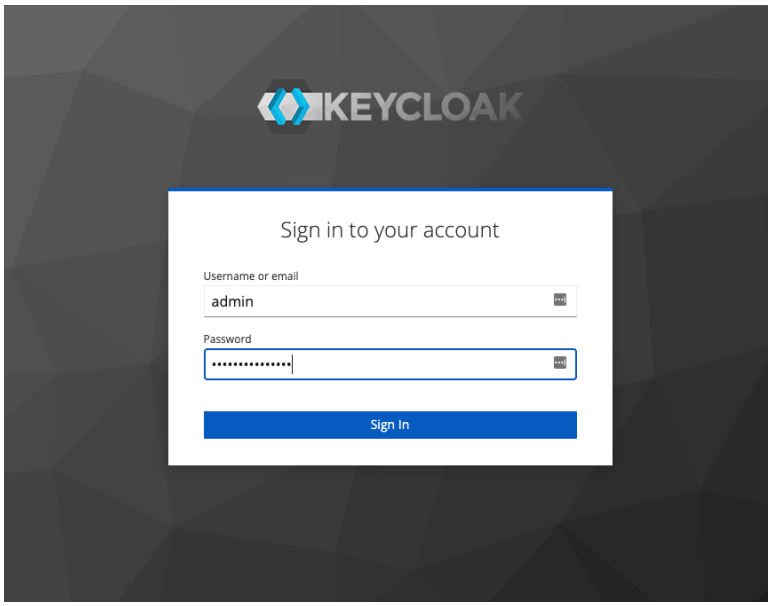
Creating User Accounts

Once you have an instance of Primer Delta Lite running, you will need to create one or more user accounts that can login to the application. User accounts are managed in the included Keycloak IAM service. A single Keycloak administrator account is created during installation. That Keycloak administrator account can be used to create Delta application user accounts or further configure IAM for your environment.

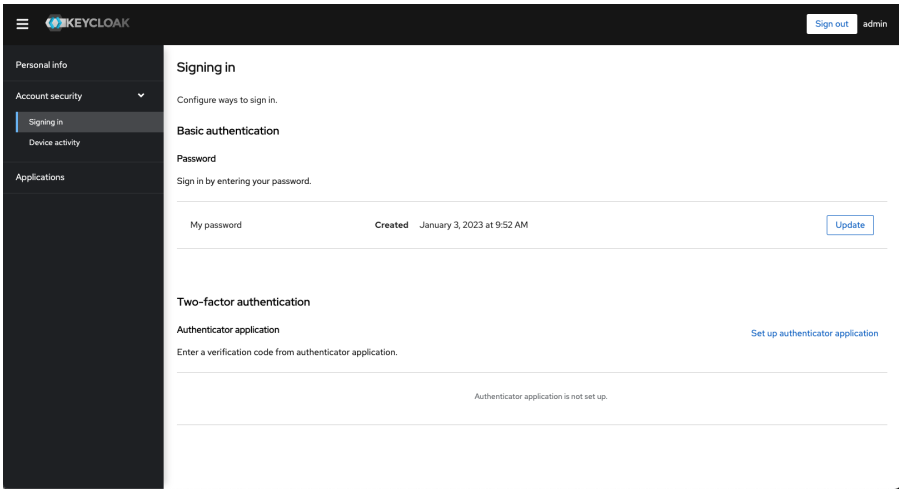
1. Change the default administrator password. Click the link given for **ChangeAdminPasswordURL** in the stack Outputs tab. You will likely be presented with a browser warning, because new instances will use a self-signed certificate by default. Select **Advanced** and **Proceed…**(or your browser's equivalent)



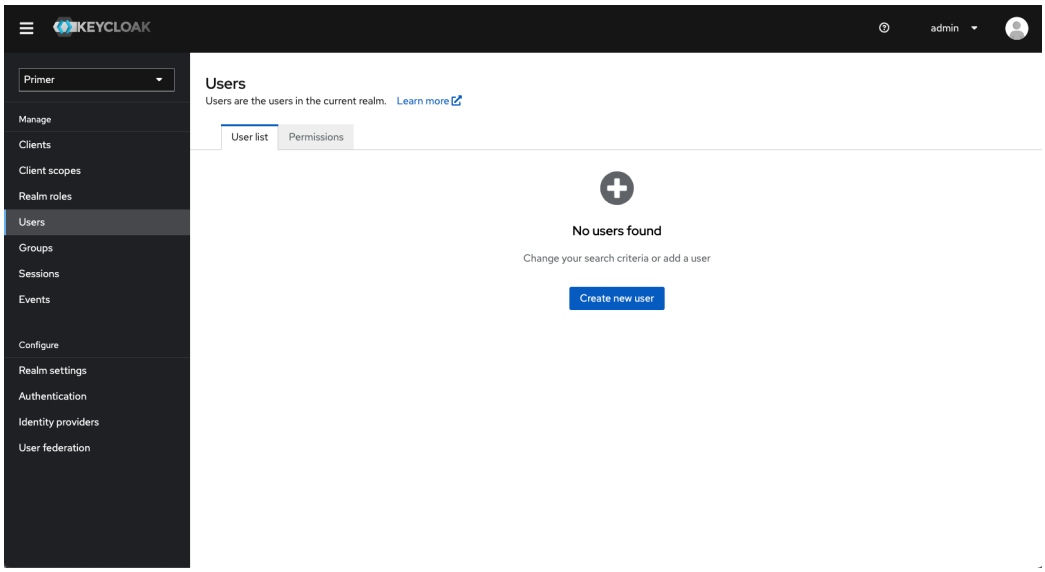
   a. Login as the administrator. The username is *admin* and the default password is *admin*



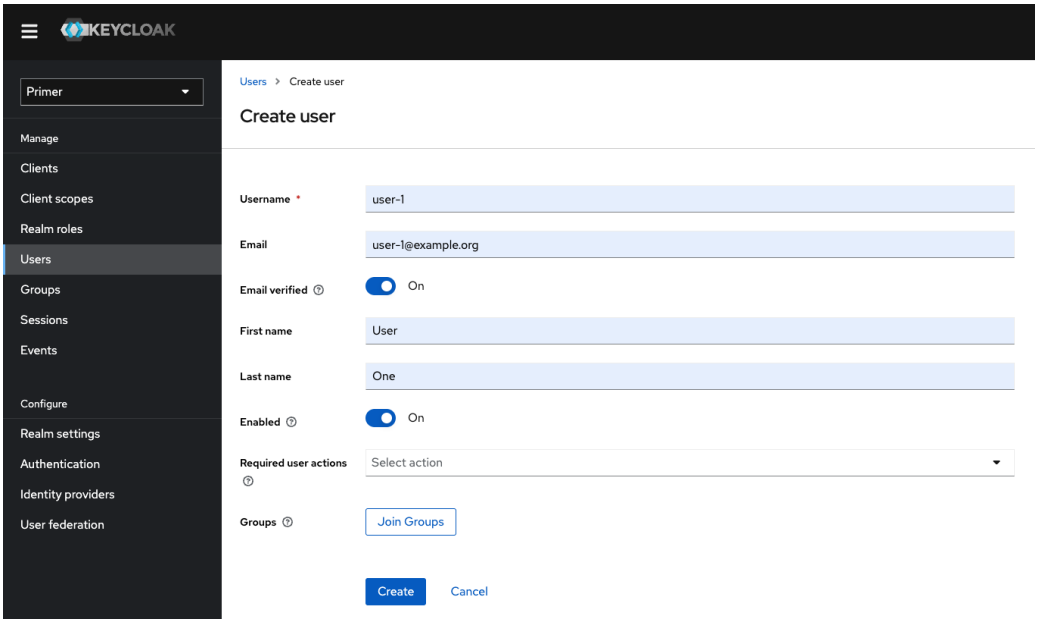
b. Click **Update** and enter a new password for the admin user.



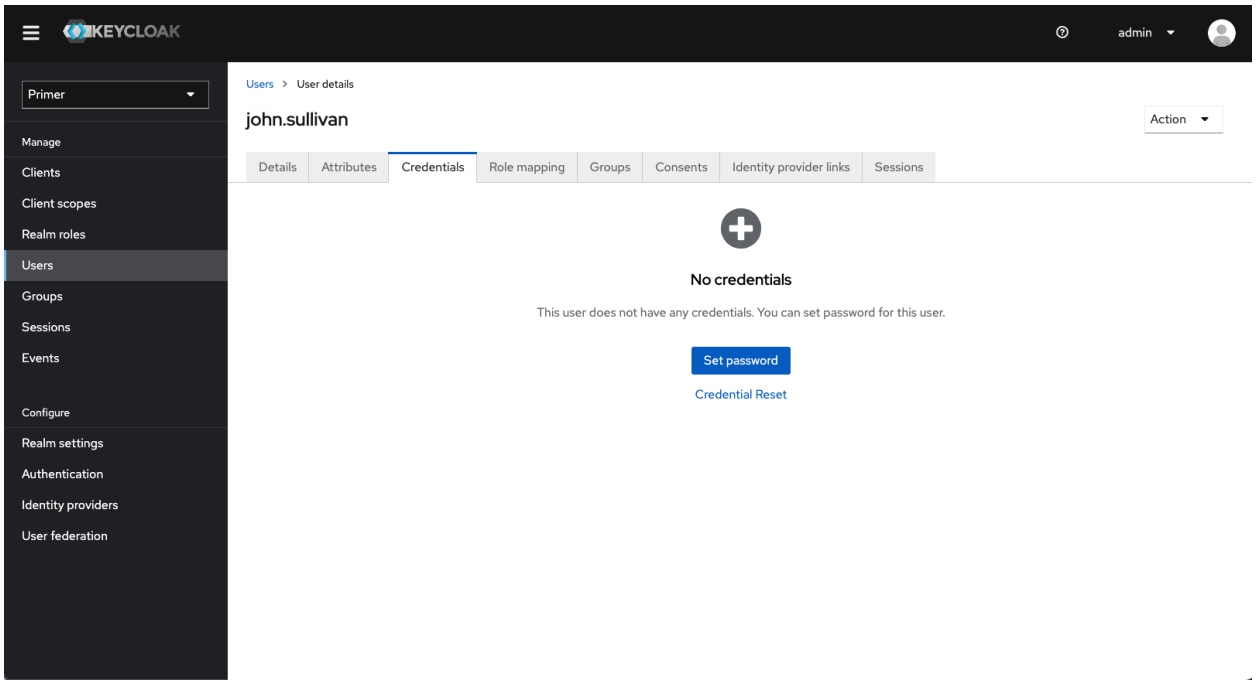
2. Before accessing the Analyze application, you must login as admin and set up one or more Delta user account(s). Click the link for **UserAdminURL** provided on the stack **Outputs** tab, login as admin, and select **Create new user**



3. Enter the username, email, first name, last name, and click **Create**



4. On the following page, select the **Credentials** tab, and Click **Set password**



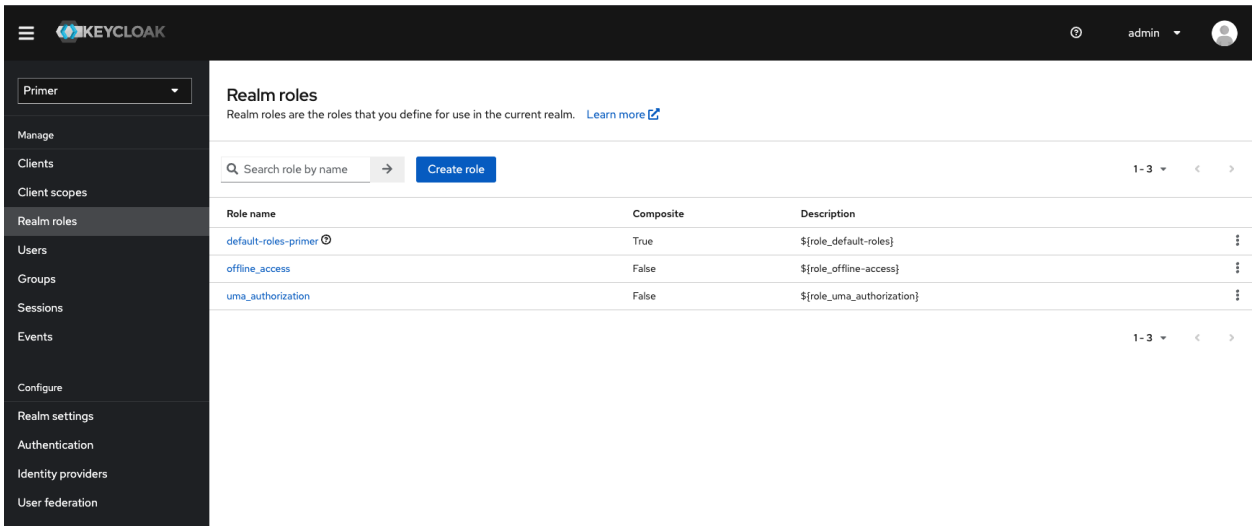
5. Enter a valid password and click **Save**. If **Temporary** is **On**, the user will be required to change their password at first login.

   Valid passwords must:

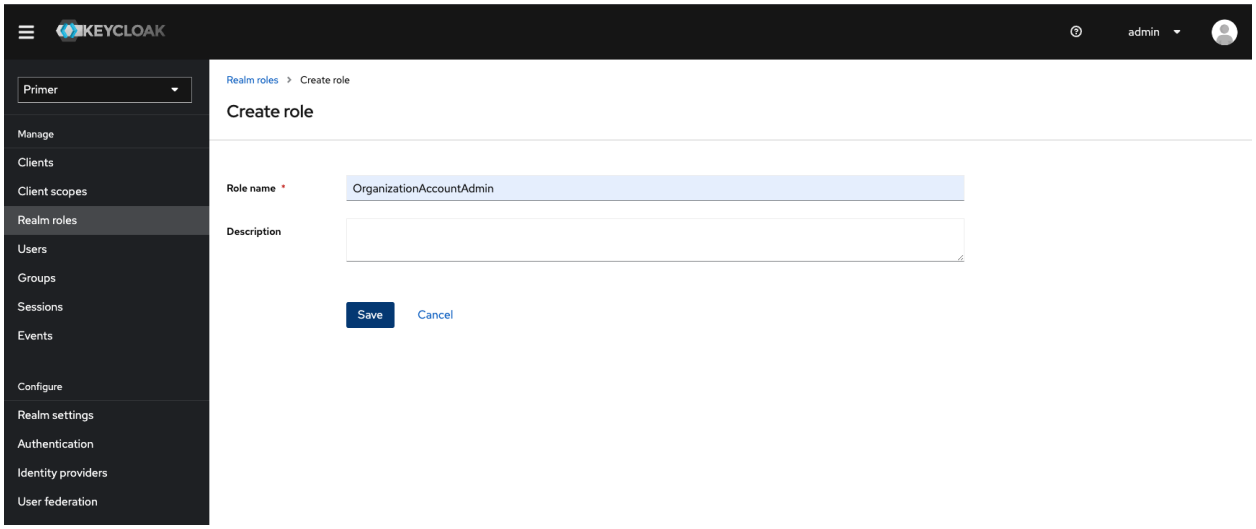
   - Be at least 15 characters
   - Have at least 1 digit
   - Have at least 1 special character
   - Have at least 1 lowercase character
   - Have at least 1 uppercase character
   - Not repeat a character more than once in a row
   - Not be one of the 7 prior passwords

6. Promote at least one user to be an administrator of the Delta application

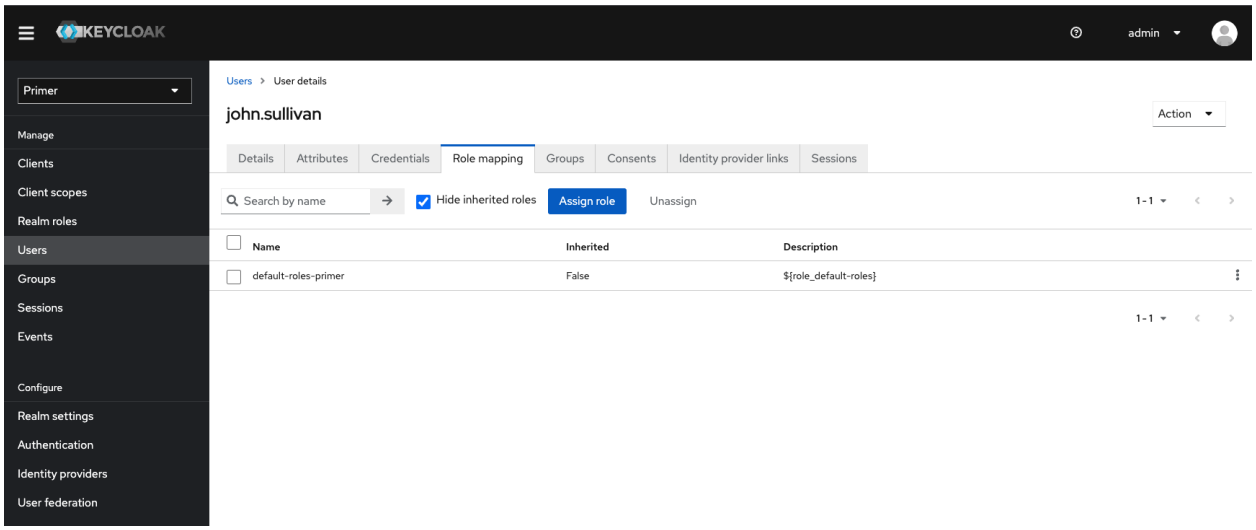
   a. Create the admin role in keycloak. From the left menu in keycloak, select **Realm roles**



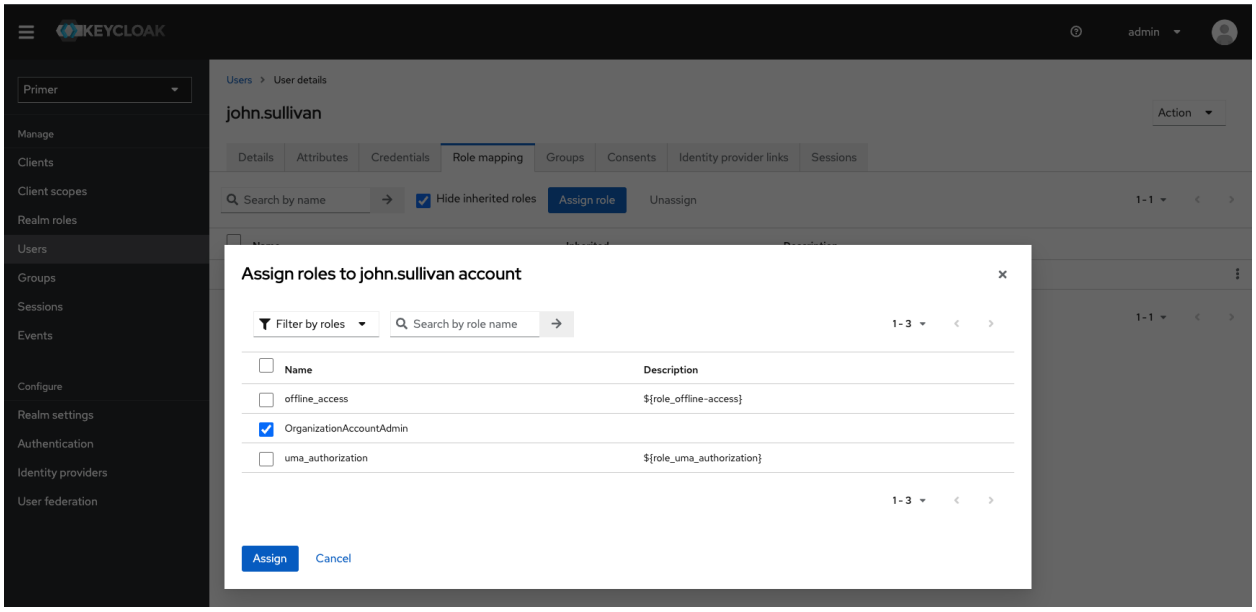
b. Select **Create role**, and enter *OrganizationAccountAdmin* as the **Role name**, and click **Save**



c. Navigate to the user you want to promote to admin. Select **Users** from the left menu, select the username, and select the **Role mapping** tab



d. Click **Assign role**, select *OrganizationAccountAdmin*, and click **Assign**



e. If the user is currently logged in to the application, they must log out before the change takes effect

7. Provide all users the **ApplicationURL** where they can access the application, along with the username and password created for them. If the password was set to Temporary, the user will be prompted to change it at their first login.

8. In order to clear warning messages in the application concerning credits and default exclusions, log in as a user with the *OrganizationAccountAdmin* role, click the gear icon, select Settings, and then save defaults for both credits and exclusions.

Importing your data to Primer Delta Lite

The fundamental data unit in Delta is the document. A document is composed of several elements, such as a title, a date, and a body of text. While Primer Delta can be integrated with a variety of data sources and data schema, Primer Delta Lite from the Marketplace is configured to work solely with data loaded into its included Elasticsearch index.

For Primer Delta Lite, a document is essentially a single document indexed in Elasticsearch and conforming to our specific JSON data format. Primer Delta Lite offers a couple of methods to bring your data to Delta and have it index that data in its Elasticsearch.

Importing Files

When you create a stack for Primer Delta Lite, it will create an S3 bucket. That bucket can be used to import your data in the form of files. Primer Delta Lite will automatically import any files that are uploaded or copied to specific folders inside that bucket.

The bucket may be accessed via the **S3IngestBucketURL** on the stack **Outputs** tab or by **IngestBucket** on the **Resources** tab.

By default, the S3 bucket will be accessible for read and write operations only by the AWS account that created the stack and the EC2 instance created for the stack. You can use S3 Bucket policy and AWS IAM to grant access to other entities. For example, you may want to

- Enable individual users to upload new files for import
- Permit a Lambda function or other AWS resource to write to the bucket as part of an automated data flow into to Primer Delta Lite
- Allow individual users to download files but not upload

The IngestBucket will have the following folder structure:

`/formatted`	Used for automatic import of JSON files formatted strictly to the Primer JSON data format
`/raw`	Used for automatic import of files of arbitrary type
`/samples`	Contains sample data files that can imported for demonstration or testing purposes

`/job-results`	Used for caching the results of analysis jobs. This folder will appear after the first analysis has completed. There is no need to work directly with the files located in this folder.

/formatted - Primer Formatted Data

You may copy or upload files with the extension `.json` or `.jsonl`. Files with other extensions will be ignored.

You may upload folders, and the resulting file structure under `/formatted` is not important.

A `.json` file must contain a single JSON document, formatted according to Primer JSON data format.

A `.jsonl` file must be a JSON Lines text document, i.e. one JSON document per line. Each line must be a single JSON document, formatted according to Primer JSON data format.

Primer Delta Lite will typically import JSON data at a rate of several hundred documents per second.

/raw - Raw File Import

Files copied or uploaded under `/raw` can be of nearly any file type. The system will make its best effort to extract text from any file. It will also look for any metadata, dependent on file format, for information like date, summary, title, publication.

PDF files that are image heavy, such as PDFs resulting from document scanning, will have OCR applied during import to extract more usable text. OCR is relatively computationally expensive, so such files may take a minute or more to import depending on the number of pages, compared to a second or less for a regular PDF or other office document formats.

You may upload folders, and the resulting file structure under `/raw` is not important.

Because Delta operates on natural language, only files that contain a reasonable amount of narrative text will be useful or interesting after import. You can upload imagery, video, or audio files, but the only resulting text will be that found in such a file's metadata, typically limiting their value at the cost of time spent reading large files.

For raw file import, the `external_url` field will always point to the location of the file in S3.

Streaming Data

The alternative to file based import is streaming data over an SQS queue. The stack will have created a queue for that purpose. The queue is identified on the stack **Outputs** tab as **SQSIngestQueueURL** and on the stack Resources tab as **IngestDataStreamQueue**. To push data via the queue, simply publish to the queue a message as a single JSON document formatted according to Primer JSON data format.

Sample Data

Sample data is provided in the bucket under the `/samples` folder. There is a single `.jsonl` file containing one day of news data (01/07/2023-01/08/2023) This can be downloaded and then uploaded to the formatted/ folder.

Alternatively, you may directly copy a sample file to the `/formatted` folder in S3.

1. From the `/samples` folder, select the file, click **Object Actions** and then **Copy** from the drop down
2. Next to the **Destination** text box, click **Browse S3**
3. Click the bucket name at the top of the dialog, click the `/formatted` folder, and then **Choose destination**
4. Click **Copy** at the bottom of the page

Managing Primer Delta Lite

Installing a Server Certificate

Obtain a valid server certificate from a trusted Certificate Authority. The certificate file should be in PEM format, and both the certificate data and the private key should be written to the same file. If you are using a registered DNS domain name to access the application, the Subject and Subject Alternate Names (SAN) for the certificate must contain a matching entry to that domain name. Similarly, if you are using the public IP address to access the application, the SAN and Subject must reference that IP address.

To install the server certificate file:

1. Copy the server certificate file to the running EC2 Instance, e.g. by using scp from a source location that is within the IP Address Range configured for SSH at stack creation.

```
scp -i <aws_keypair> mycert.pem  ec2-user@<PublicIpAddress>:/home/ec2-user/
```

2. Connect to the running EC2 instance via SSH

```
ssh -i <aws_keypair> mycert.pem  ec2-user@<PublicIpAddress>
```

3. Using sudo, copy the server certificate file to `/etc/haproxy/haproxy.pem`

```
sudo cp mycert.pem /etc/haproxy/haproxy.pem
```

4. Ensure the file has appropriate permissions

```
sudo chmod 600 /etc/haproxy/haproxy.pem
```

5. Restart the haproxy service

```
sudo systemctl restart haproxy
```

Update the SSH IP Address Range

You may need to update the SSH IP Address Range at some point to maintain SSH access to your EC2 instance. The IP address based access control is implemented as a Security Group associated with the EC2 instance. To modify the security group:

1. Navigate to the AWS CloudFormation console and select the stack created for Primer Delta Lite
2. Select the **Resources** tab and then click on the link provided for **InstanceSecurityGroup** to open the EC2 Console with the security group selected
3. Select the **Inbound Rules** tab and click **Edit Inbound Rules**
4. Edit the existing rule for TCP 22, or add a new rule for TCP 22, as appropriate
5. Click **Save Rules**

Managing Data

Data imported into Delta is stored in an Elasticsearch index. If you want to manage that data, the Elasticsearch API is available from the command line of the running EC2 instance. For instance, you can SSH to the EC2 instance and use curl commands to operate on Elasticsearch.

A common use case would be to delete data that you no longer want for analysis. For example, if you loaded the sample data included in the S3 bucket to try out the application, you may want to delete it before importing your own data. Or you may have exported your data to the Primer JSON format and, after importing to Delta, decided to change your export logic and now want to re-import the data.

To delete all data in the Delta document index, use the following command after you SSH to the EC2 instance:

```
curl -X DELETE http://localhost:9200/canonical_docs
```

To be more selective, you may instead delete by query. For example, to delete all documents from the dates of the sample news data, use the following command:

```
curl -X POST http://localhost:9200/canonical_docs/_delete_by_query?pretty
-H 'Content-Type: application/json' -d'
{
    "query": {
        "range": {
            "date": {
                "gte": "2023-01-07",
                "lte": "2023-01-08"
            }
        }
    }
}'
```

Managing Users

User management in Primer Delta Lite is performed via Keycloak. Keycloak is an open-source software product that provides IAM services for applications and services. In the default configuration for Primer Delta Lite it serves as the central authentication and authorization server. As described Creating User Accounts, you can create users, assign password credentials, and grant users the application administration role. However, Keycloak is a flexible platform for IAM and you can customize it to your needs. Please reference the administrator guide for Keycloak for any customization you may want to perform.

You can give users the ability to manage their own accounts, namely change their passwords and set up MFA, via the account admin page in Keycloak for the Primer realm. To enable that functionality, authenticate to the Keycloak admin console, linked in the CloudFormation stack **Outputs** tab as

1. Open the Keycloak admin console, using **UserAdminURL** from the CloudFormation stack **Outputs** tab
2. Authenticate as the *admin* user
3. From the left menu, select **Clients**
4. Click on **account-console**
5. Select the **Client Scopes** tab
6. Click Add client scope
7. Select *roles* and click **Add**, choosing **Default**

Users can then go to https://<server address>/realms/primer/account/, where they can authenticate with their given credentials and manage their account.

The user registration flow can also be implemented as self-service. This means that, instead of the administrator creating user accounts in the Keycloak admin console, users can register themselves, setting their name, email, and password. This flow requires email integration; a user must validate their email by receiving an email sent by the Keycloak server. You can configure the email connection for the Primer realm by following the instructions provided in the Keycloak administrator guide.

You can also replace or augment the default username/password authentication with your enterprise authentication service. In addition to serving as an authentication service, Keycloak can act as an identity broker and delegate authentication and identity responsibilities. For example, you can configure the authentication flow for the Primer realm to connect to an external identity provider, integrated with Keycloak via OpenID or SAML. When a user tries to access the Delta application, Keycloak will forward them to the authentication page for your enterprise identity provider. Once authenticated, that authentication will then flow back through Keycloak and the user will be authenticated to Delta. Additionally, Keycloak can be integrated

with an external LDAP or ActiveDirectory to pull user identity, such as name, email, and roles. Please reference the Keycloak documentation for details on how to do these things.

Appendix

CloudFormation Parameters Reference

Parameter	Description
Availability Zone	The AWS availability zone (AZ) where the VPC subnets will be created. The most important factor in picking an AZ is choosing one that supports the selected EC2 Instance Type. Not all instance types are available in all AZs. If the selected AZ does not support the selected instance type, stack creation will fail when it tries to run the EC2 instance and CloudFormation will rollback the stack. Please see Troubleshoot EC2 instance Availability Zone errors at launch for more information on picking an AZ for an instance type.
EC2 Instance Type	Provides a list of supported instance types. The application will run on any of the options. The maximum number of documents that will be included in an analysis is based on the selected instance size. The maximum number is 1000 documents for EC2 instance sizes with > 20GB memory, and 500 documents for EC2 instance sizes with less memory. Larger instance types, with more CPU and memory resources are recommended when importing large and/or complex file formats. For example, if you plan to import PDF files that are document scans, those will undergo OCR processing, which is CPU and memory intensive. Choosing the larger option for an instance class, e.g. *4xlarge*, in that case will result in a noticeable performance improvement during data import. Otherwise, a smaller instance type should generally be sufficient. Not all instance types will be supported for the selected AZ (see above).
Key Pair	Provides a list of keypairs registered in the current region for your account. The person who possesses the private key for the pair can use it to ssh to the running EC2 instance.

SSH IP Address Range	This controls SSH access to the EC2 instance based on IP address. It must be a range of IPv4 addresses, in CIDR block notation. The default is 10.0.0.0/16, which will limit SSH access to sources that are inside the VPC. In practical terms, you would need to start an EC2 instance in the VPC as a “bastion” or "jumpbox” and hop from that instance to the EC2 instance hosting the application. 0.0.0.0/0 would allow access from any source. You can limit access to a single address using /32, e.g. 203.0.113.1/32. This is handy for granting access to only your client machine, using its public IP address.
HTTPS IP Address Range	This controls HTTPS access to the EC2 instance, effectively end user access to the application, based on IP address. It must be a range of IPv4 addresses, in CIDR block notation. 0.0.0.0/0 will allow access from any source. You can limit access to a single address using /32, e.g. 203.0.113.1/32.
Use a registered DNS name	Each stack will create an Elastic IP (EIP) address for the application, i.e. static public IP address. When this parameter is set to “None”, the application will be configured to use that EIP as its public address. If you plan to instead register a domain name in DNS for that EIP, you will need to provide that name here in order for the application to configure itself to use that name instead of the IP address. It is important to note that you will absolutely need to obtain a valid PKI server certificate for the registered domain name, and load that certificate in the EC2 instance. The application generates a self-signed certificate for the public address, to use as a default. Most browsers will allow a user to accept the self signed certificate for an IP address, so you can use the application in that state if you prefer.. However, most browsers will not allow a user to accept a self signed certificate for a domain name address, making the application effectively unusable until a valid server certificate is loaded.
System Control Marking	This is the security control marking that is displayed in a banner at the top of every page in the application. This may be used to indicate the system level classification or controls that are in effect for any data imported into the application. For cases where data controls are not needed, this can be used to set an arbitrary title on each page, e.g. to distinguish one instance of the application from another when running multiple stacks for different purposes.

Default Document Portion Marking	This is the security control marking to display for document content in the application, when no such security control mark was set for the document during import.

Primer JSON data format

Our JSON data format consists of the following fields. The product depends on these fields for its algorithms. While the product will function without some or even most of the fields specified, failing to provide the appropriate fields for your data will result in the loss of some features of the analysis.

Field Name	Type	Description
id	text	**Required**. The unique identifier for a document in your data. Must be unique across all documents imported to Delta.
title	text	The title of a document in your data. For example, in news data, this field would be the headline or title of a news article.
body	text	The content of a document in your data. For example, in news data, this field would be the actual text of a news article. We use this field to create topics, events, summaries, and reading lists, as well as pick out numbers, locations, people, dates, and quotes from the documents returned for a query. Ideally the body text for documents should only include content which is useful for analysis. Additional paragraphs of text, odd tags, metadata, and boilerplate text can throw off the clustering and result in less useful topics and events and cause the inclusion of unintended entities.
summary	text	A short summary of the document body. Used for display purposes.
date	datetime	The date associated with a document in your data. We use this field to generate events, prioritize documents or events by their recency, make time-related statements about the documents in the analysis, and display time series graphs throughout the product. Depending on the exact use case you may want to use either the date of information or date of publication in this field. The format is `yyyy-MM-dd'T'HH:mm:ss.SSSXXX`
publication	text	The publisher of a document in your data. For example, in news data, this field would be the name of the publication in (or site on) which an article appeared. This data is shown in document tables, on the publications tab, and when presenting links to documents.
title_pm	text	The security control marking to display along with the document title, indicating any controls to be applied to the document as a whole.

Field Name	Type	Description
document_pm	text	The security control marking to display for portions of the document when presented on their own.
external_url	text	A URL that links back to the original document or the system where the document may be accessed. Users will be given this link in order to navigate back to the original source.

Sample JSON Document


```
{
  "id": "655f00de",
  "title": "REPORT",
  "body": "On 06JAN12, Sector Puget Sound received NRC 1070247 reporting a
discharge of hydraulic fluid in Elliott Bay, in Seattle, WA. The F/V OCEAN
ROVER was using one of its cranes to move a skiff when a hydraulic line
coupling came undone, causing the charged line to discharge approximately 2
cups of hydraulic fluid into Elliott Bay, a U.S. navigable waterway. The
discharged fluid created sheen upon the water. The line was repaired and
absorbents pads were applied to the water to soak up the remaining
hydraulic fluid sheen.",
  "summary": "discharge of hydraulic fluid",
  "date": "2012-01-06T00:00:00.000Z",
  "publication": "Coast Guard Watch Logs",
  "document_pm": "(U)",
  "title_pm": "(U)",
  "external_url": "http://catalog.data.gov/dataset/maritime/log/655f00de"
}
```

CloudFormation Stack Least Privilege Policy

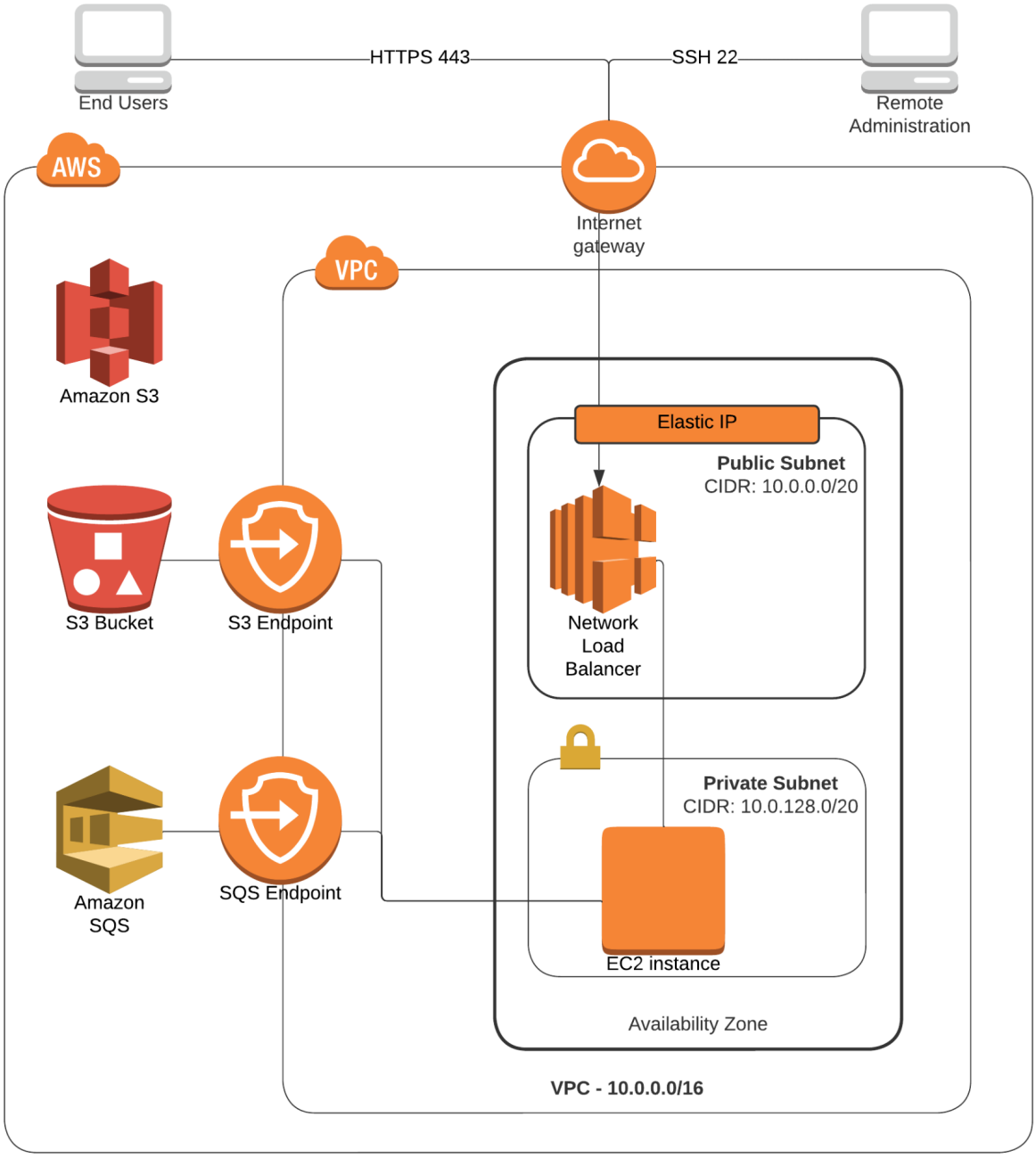
The following JSON is a valid AWS IAM Policy that defines the minimal permissions necessary to create and delete the CloudFormation stack for Primer Delta Lite.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AcceptAddressTransfer",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateInternetGateway",
                "ec2:CreateNetworkAcl",
                "ec2:CreateNetworkAclEntry",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNetworkAcl",
                "ec2:DeleteNetworkAclEntry",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteSubnet",
                "ec2:DeleteTags",
                "ec2:DeleteVpc",
                "ec2:DeleteVpcEndpoints",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateAddress",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:ReleaseAddress",
                "ec2:ReplaceNetworkAclAssociation",
                "ec2:ReplaceRouteTableAssociation",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "elasticloadbalancing:AddTags",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetHealth",
                "elasticloadbalancing:RegisterTargets",
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:CreateRole",
                "iam:DeleteInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:GetRolePolicy",
```

```
                "iam:PutRolePolicy",
                "iam:RemoveRoleFromInstanceProfile",
                "s3:CreateAccessPoint",
                "s3:CreateBucket",
                "s3:DeleteAccessPoint",
                "s3:DeleteBucket",
                "s3:DeleteAccessPointPolicy",
                "s3:PutAccessPointPolicy",
                "s3:PutAccessPointPublicAccessBlock",
                "s3:PutBucketNotification",
                "s3:PutBucketTagging",
                "s3:PutEncryptionConfiguration",
                "sqs:AddPermission",
                "sqs:CreateQueue",
                "sqs:DeleteQueue",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:SetQueueAttributes",
                "sqs:TagQueue",
                "sqs:UntagQueue"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ec2.amazonaws.com"
                }
            }
        }
    ]
}
```

CloudFormation Stack Architecture



Secure Configuration

Primer Delta Lite is built and configured to comply with US Government security standards. The EC2 instance operating system is Red Hat Enterprise Linux 8, configured to run with the DoD STIG profile. All Primer developed software undergoes continuous source code scanning and vulnerability scanning. All major third party software services used in the solution are sourced from the DoD IronBank container repository.

All application processes are run with non-root permissions, with the exception of the HAProxy reverse proxy, which runs as root in order to bind to port 443. Ports exposed from the EC2 instance are limited to 443 and 22. Ports are enforced by both the OS firewall and the EC2 Instance Security group. The EC2 instance runs in a private subnet and all traffic to the EC2 instance is routed through AWS Elastic Load Balancing. There is no outbound traffic from the EC2 instance, other than responding to 442 and 22 connections. SSH access is controlled by IP address range via a Security Group; the default range allows only traffic from within the stack created VPC.