Amazon SQS Setup Guide

Nayax | amazon SQS

Version: 0.3 Date Produced: 27.04.2024 Produced by: Nayax

Document Control

Version	Author	Date	Description	Approved by
01	Oleg Rozenblum	31/03/2020	Initial Release.	
02	Oleg Rozenblum	31/03/2020	Added Fields to Transaction Dispatch	
03	Oleg Rozenblum	27/04/2020	Updated Field types and Descriptions	

Table of Content

Document Control 02
Table of Content 03
Overview 05
Definitions 05
Amazon Queue Setup 06
Creating a New Queue 06
Select AWS Region 06
Create Queue 07
Find Queue to ge URL and ARN 08
Queue Permissions Setup 09
IAM - Manage Access to AWS Resources 09
Create Policy 10
Choose Permissions 11
Set Policy for Specific ARN 11
Create User for Queue Access 14
Add User 14
Set Permissions 15
Add tags 16
Save Access aand Secret Key 19
Nayax Back Office Setup 19
Transaction Delivery Setup 19
Operator Level Setup 19
Example Transaction JSON Message 20
JSON Message Fields Description 21

EVA DTS DEX/DDCMP Audit file Delivery Setup **25**

Operator Level Setup **25**

Machine Level Setup **26**

Example DEX/DDCMP Audit file JSON Message **27**

JSON Message Fields Description **27**

Events Delivery Setup **28**

Operator Level Setup **28**

Example Event/Alert JSON Message **29**

JSON Message Fields Description **29**

Setup Encryption on Amazon SQS Delivery **30**

Generate Encryption Keys **30**

List Encryption Keys **32**

Overview

Amazon SQS (Simple Queuing Service) is a Queue Mechanism that allows a robust, redundant and reliable way to transfer data between systems without the hassle of managing retry mechanisms, Nayax has created a mechanism to deliver key pieces of information like Transactions, EVA DTS Audit files and Events to Amazon SQS Queues, this document will outline the setup of this mechanism in Nayax back end and required steps on Amazon Web Services management console

Definitions

Description	Approved by
Amazon SQS	Amazon Simple Queue Service, Fully managed message queues for microservices, distributed systems, and serverless applications
EVA DTS	European Vending Association Data Transfer Standard
DEX / DDCMP	Audit File Containing Vending Machine Counters as defined in the EVA DTS
JSON	JavaScript Object Notation, an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format, with a diverse range of applications, such as serving as replacement for XML in AJAX systems.
ARN	Amazon Resource Name
Access Key	Amazon SQS Queue Access Key
Secret Key	Amazon SQS Queue Access Key
Queue URL	Amazon SQS Queue URL

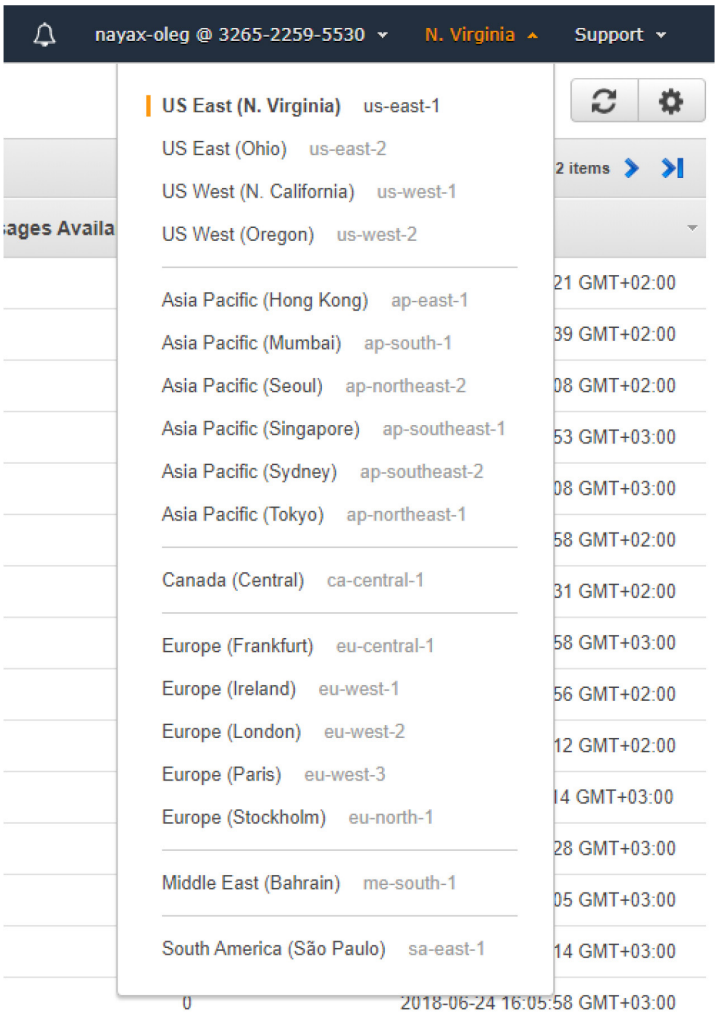
Amazong Queue Setup

Create a New Queue

To create a new Queue you will need to create an Amazon Account, afterwards login to AWS Management console and follow the below described steps

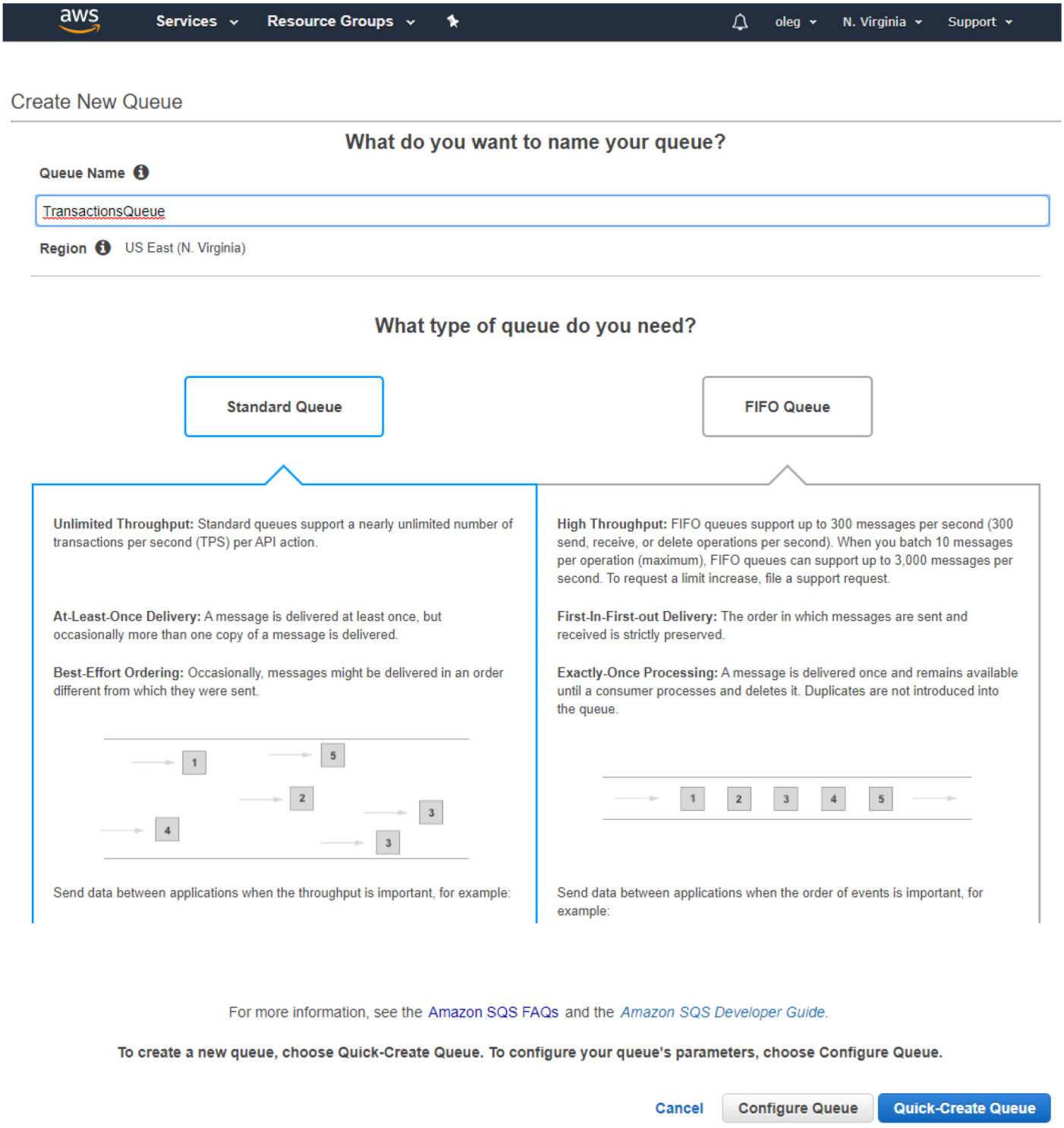
Select AWS Region

In the top right corner, choose region where you will create the queue, region selection affects AWS pricing and latency



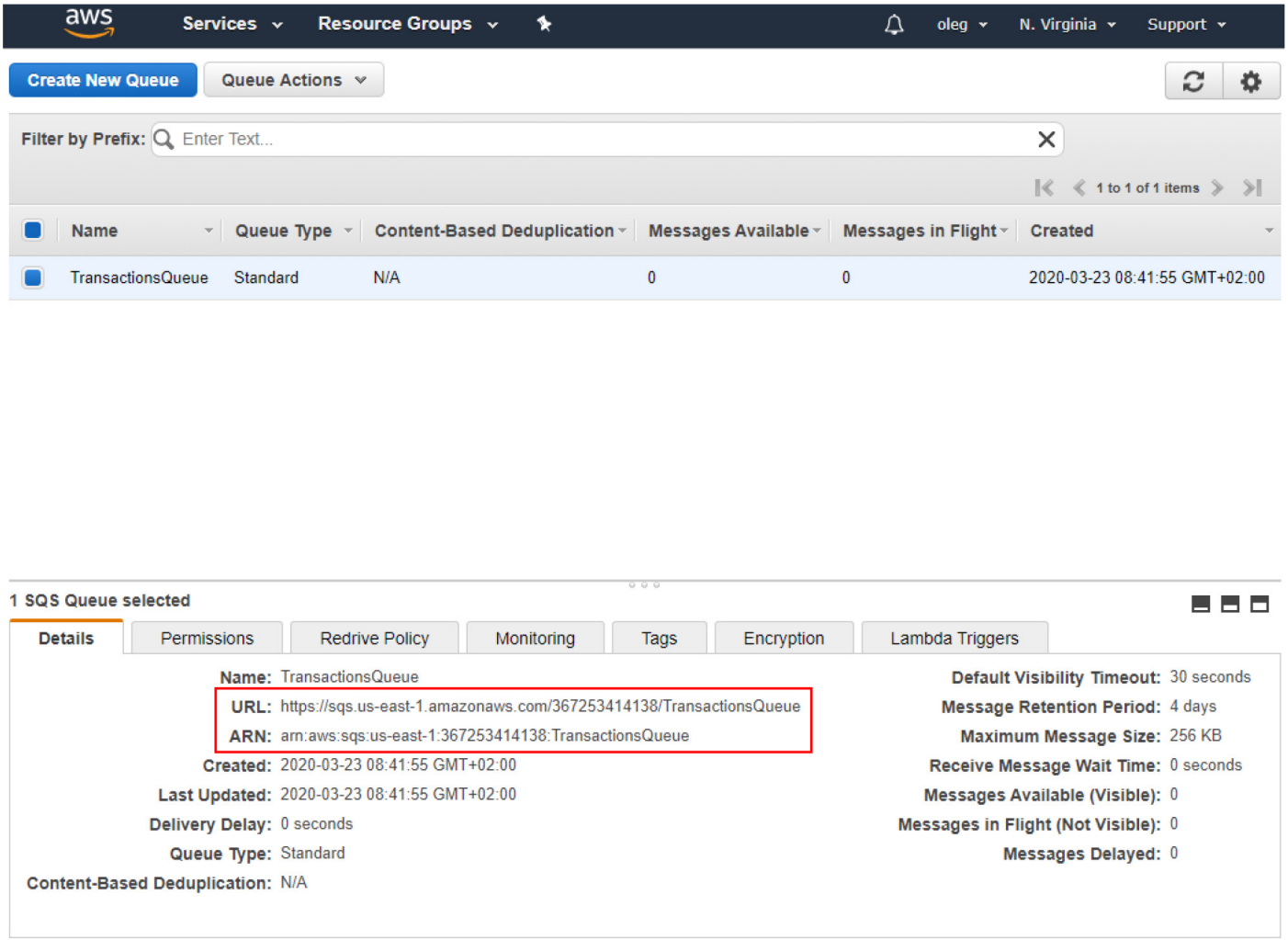
Create Queue

Create New Queue, set Queue Name, choose **Standard Queue** and press **Quick-Create Queue**



Find Queue to get URL and ARN

Once Queue is created, select the queue to find **Queue URL** (part of Queue Credentials) and **ARN** (used for Permissions setup)



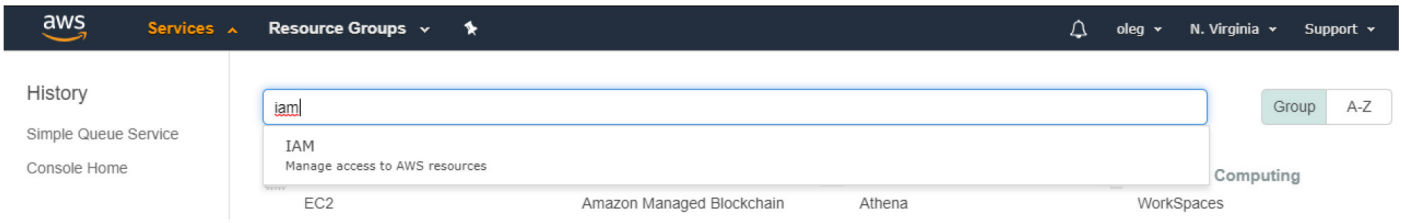
Copy the ARN to side note as you will need it for the next step of Policy setup

Queue Permissions Setup

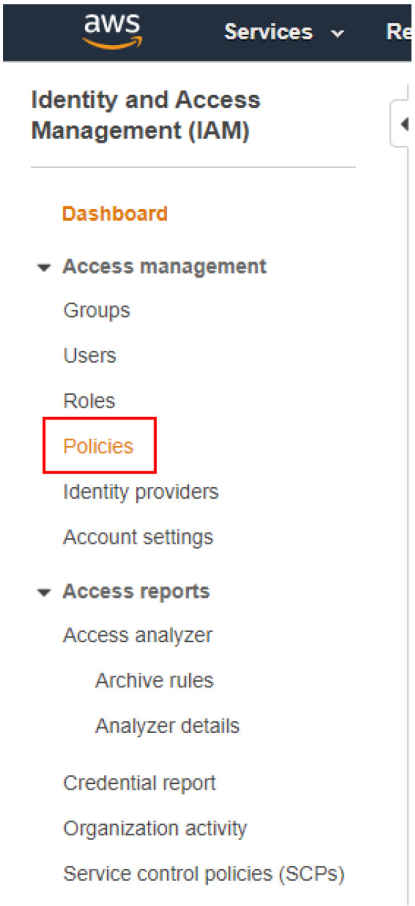
In order to manage access to Queue you will need to create a policy, then user access and assign the policy to the created user, Policy contains permissions set that define type of actions allowed on the queue

IAM – Manage Access to AWS resources

Under Services, Search for IAM (Identity and Access Management), and click on it

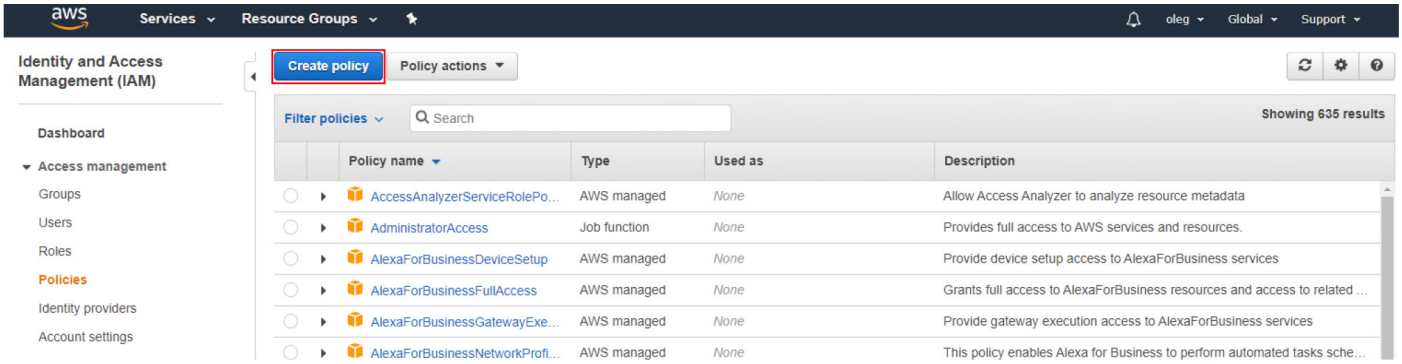


In the left side menu, choose Policies

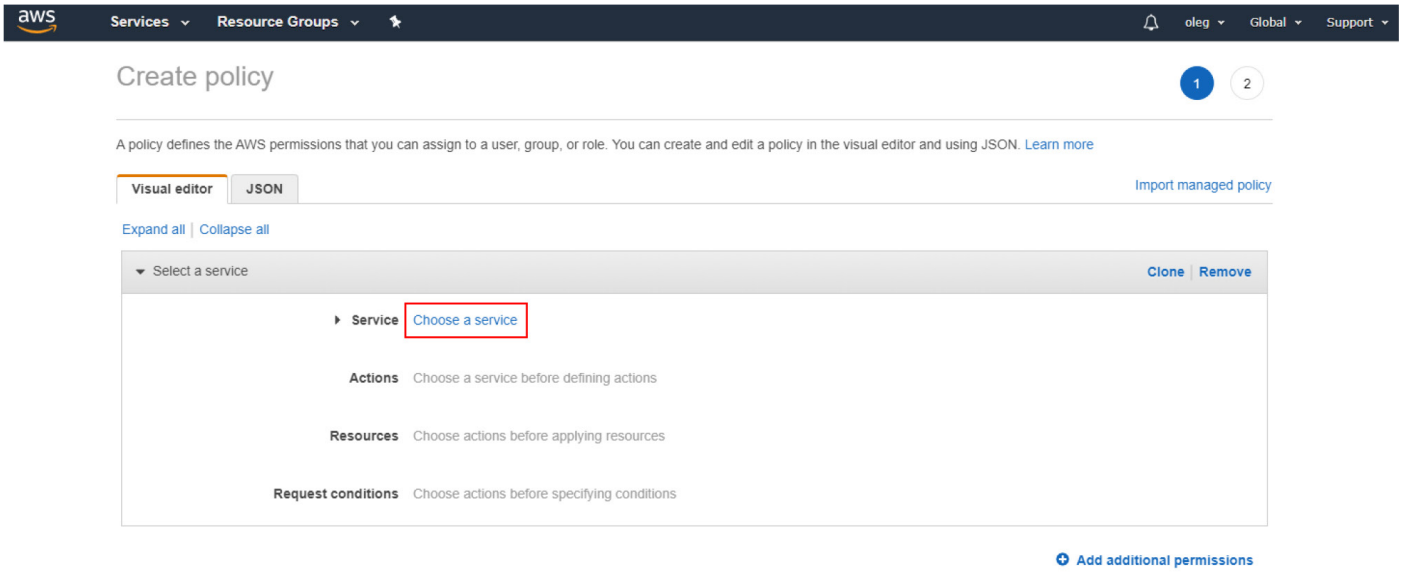


Create Policy

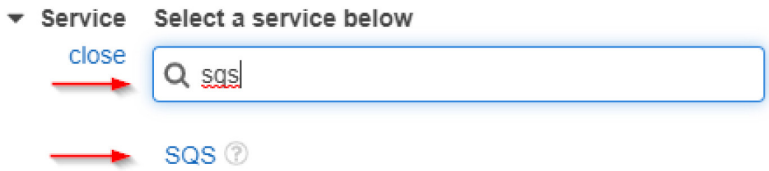
Click on Create Policy



Under Service click on "Choose a service"

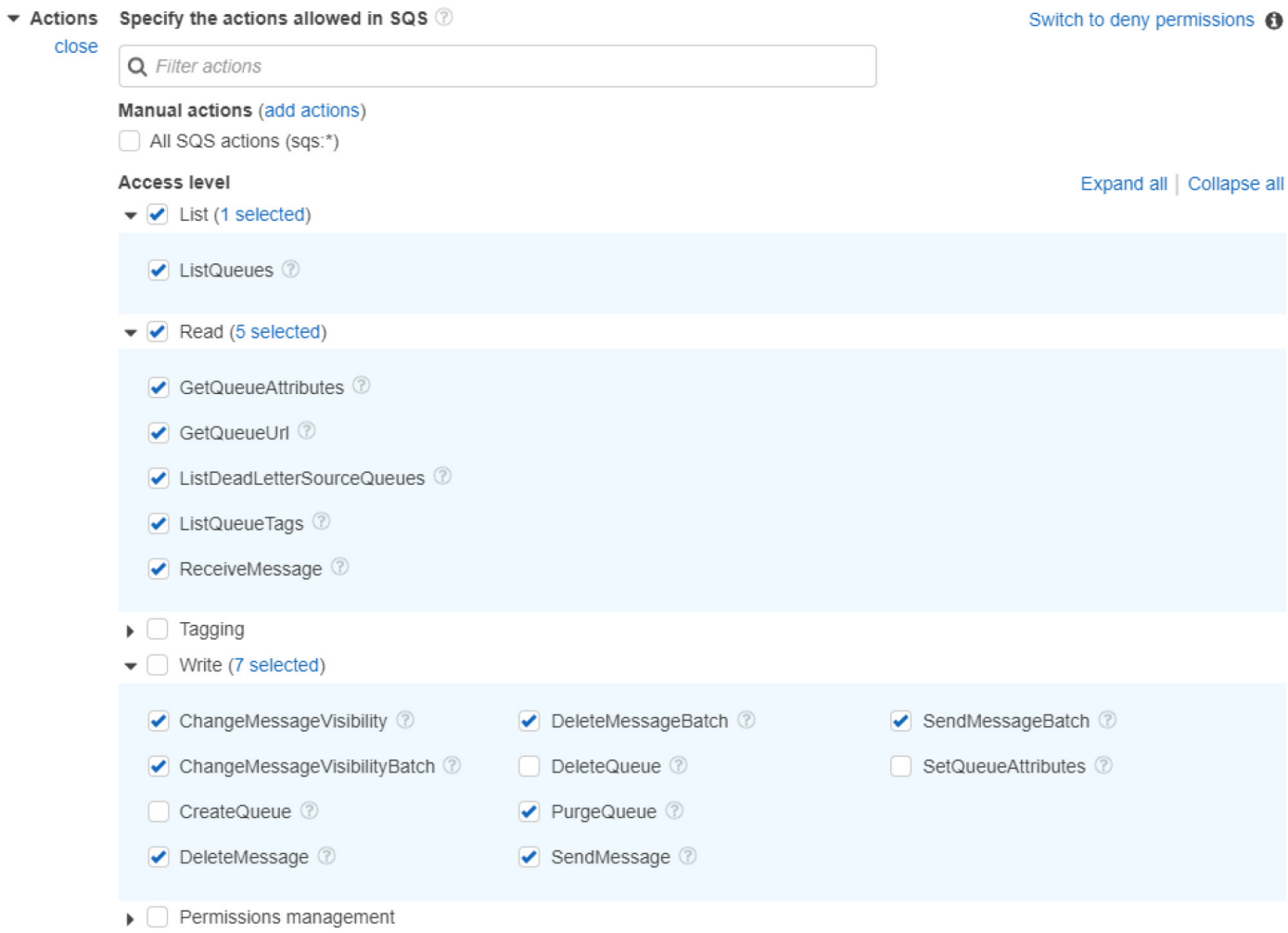


Search and Select SQS



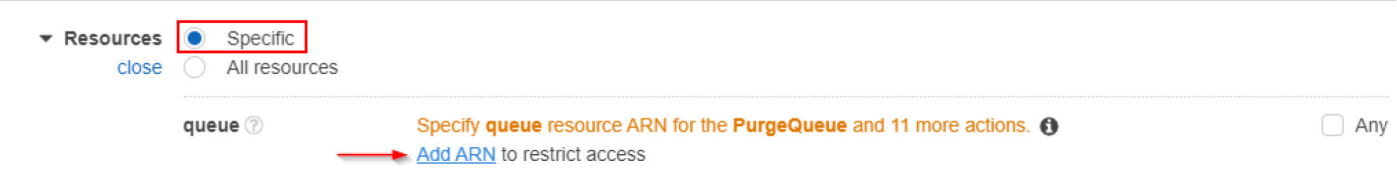
Choose Permissions

Under Actions Choose the following permissions



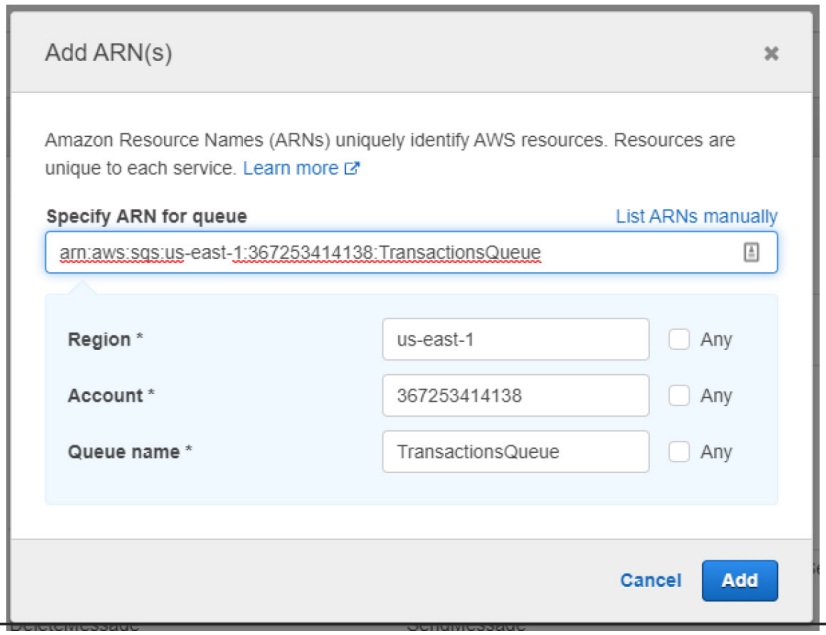
Set Policy for Specific ARN

The Policy we are creating can be assigned to any Queue,
in our case we will assign it to a specific Queue

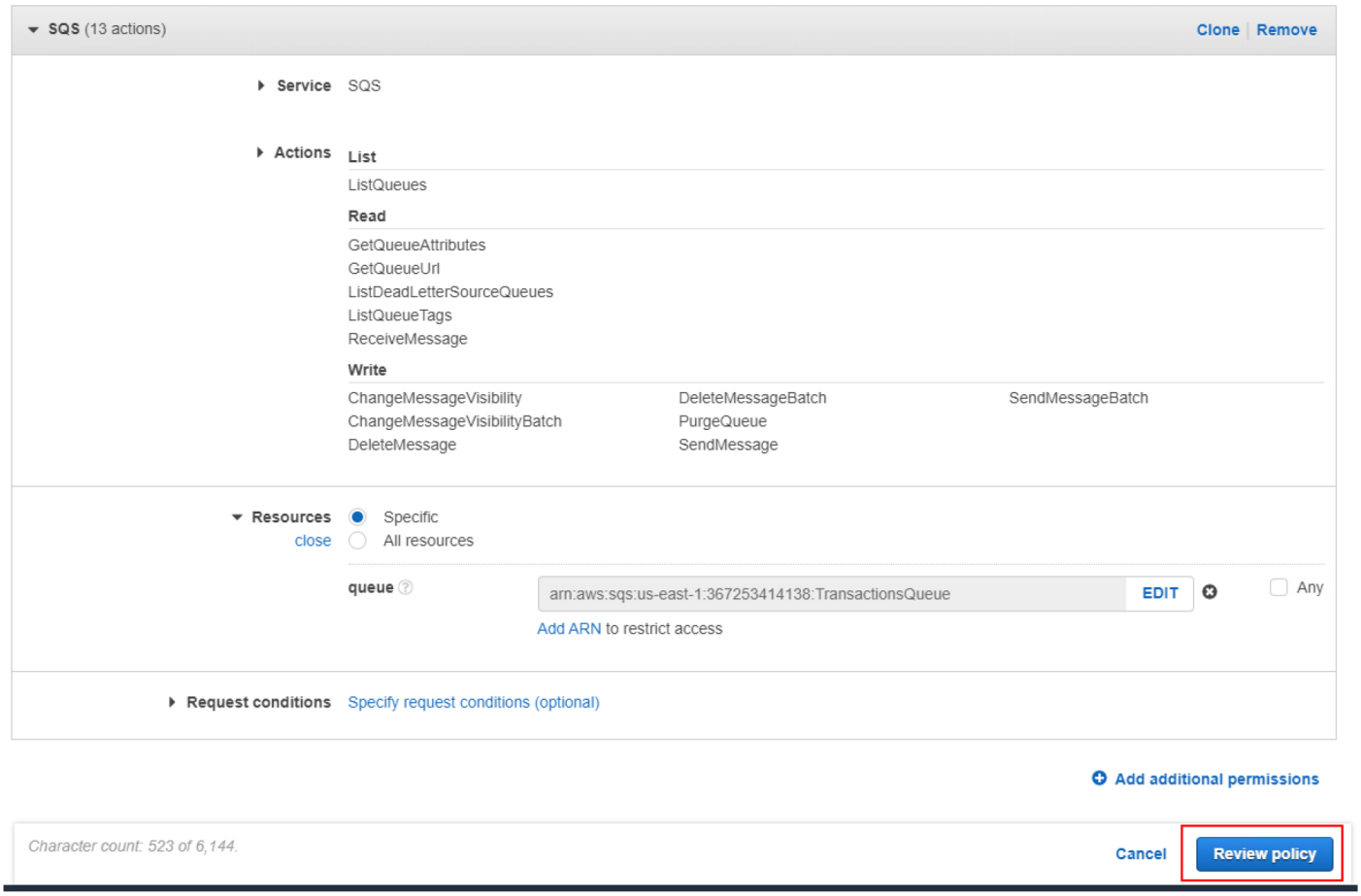


For this step you will need the ARN of the Queue you have created

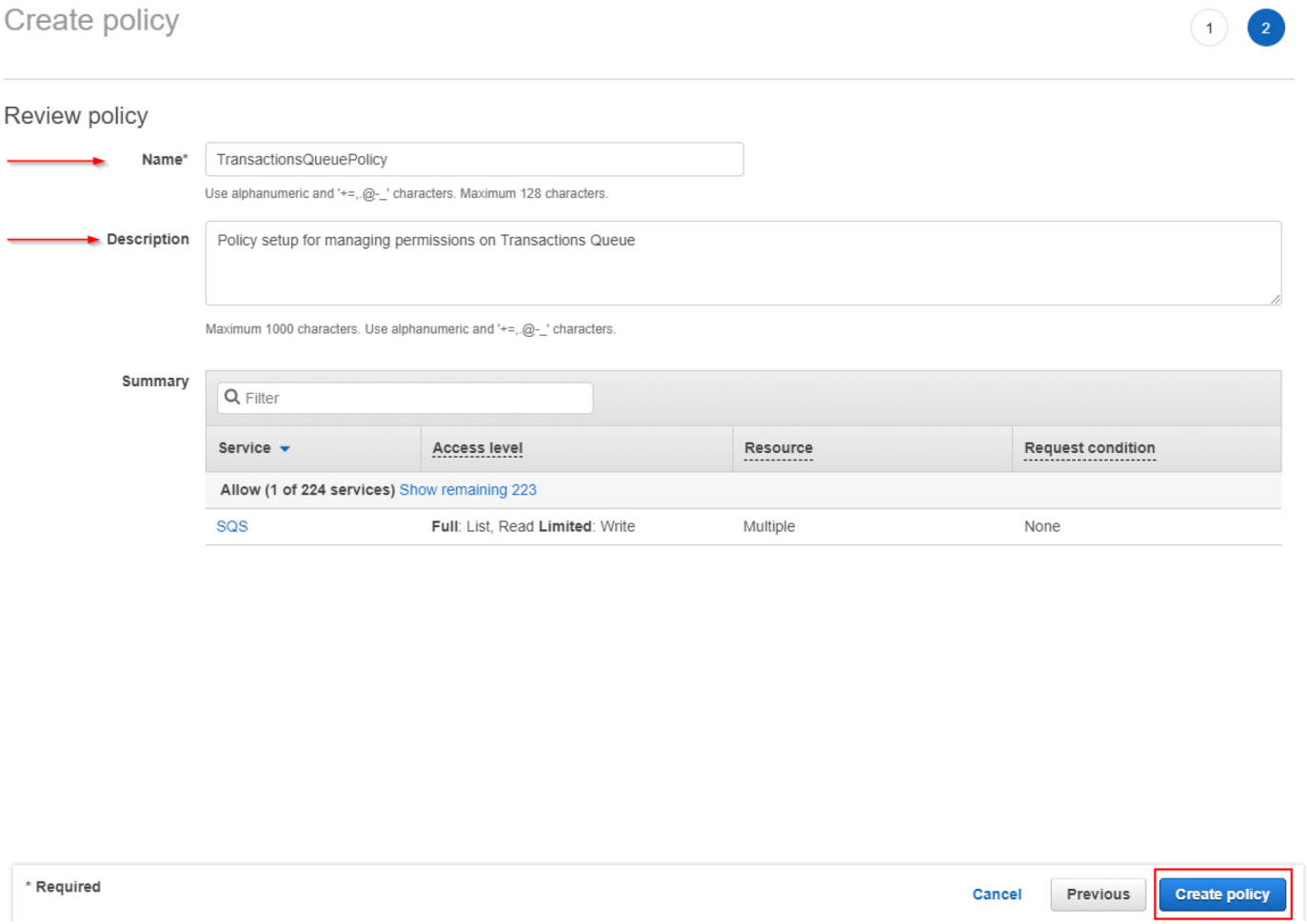
Once you click "Add ARN" a popup window will appear where you will need to paste the copied ARN and click "Add"



Then Click on Review Policy



Give the Policy a Name and Description(optional) and click "Create Policy"



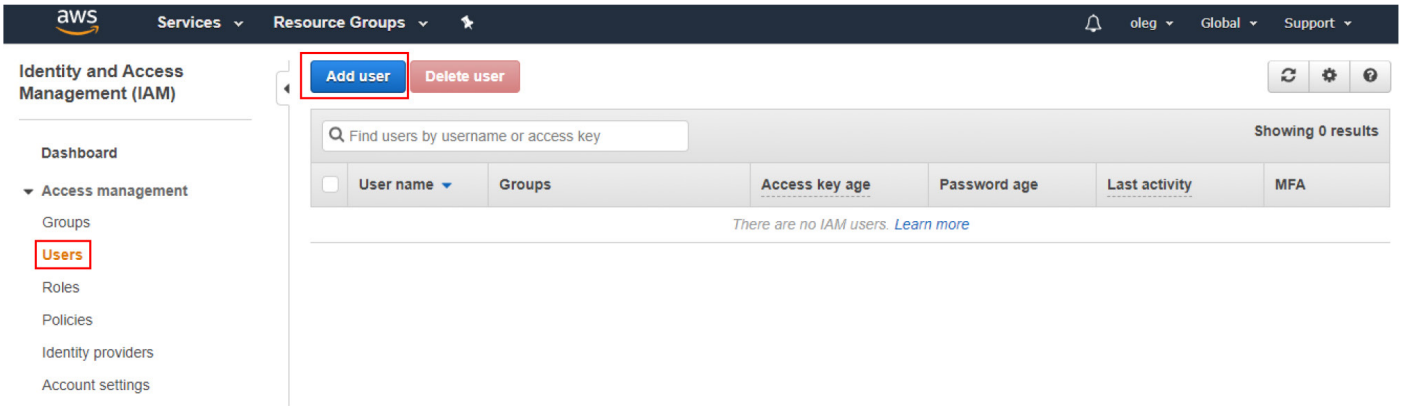
You should get an indication that Policy has been created

Create User for Queue Access

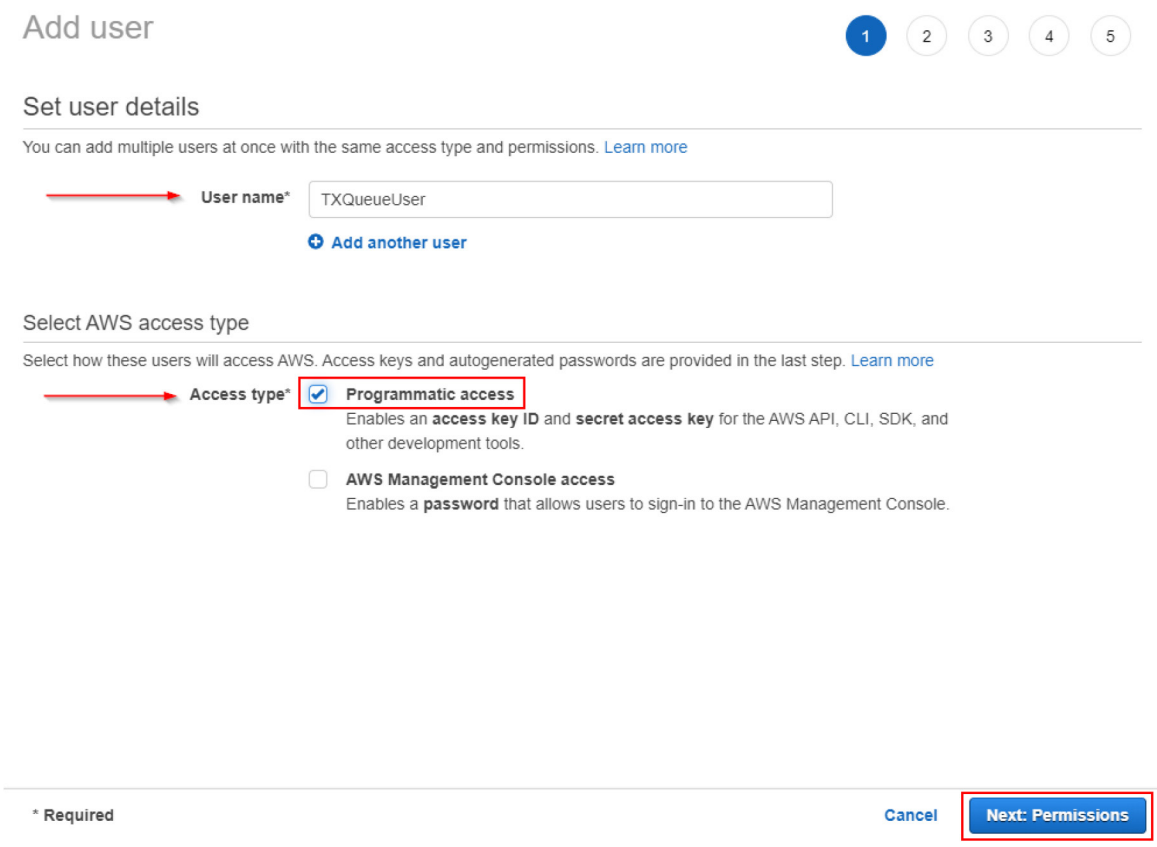
In order to access the queue and perform actions on it you need to create a User, there are 2 types of users that can be created, Programmatic Access or AWS Console Management Access, in our case we are going to create a Programmatic Access user

Add User

In the left side menu under Identity and Access Management (IAM) click on Users and Add User button

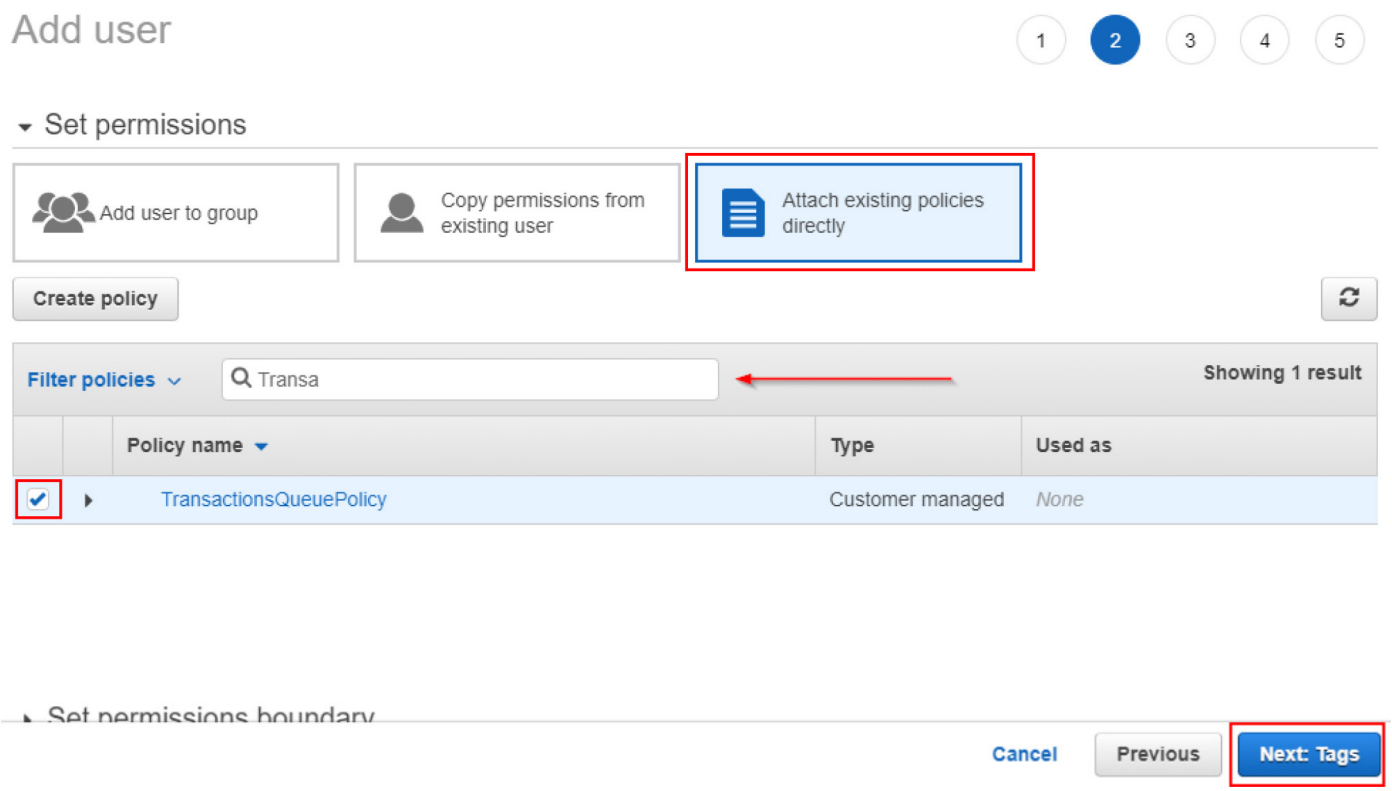


Set Username, Select Access Type – Programmatic Access and Click "Next: Permissions" button



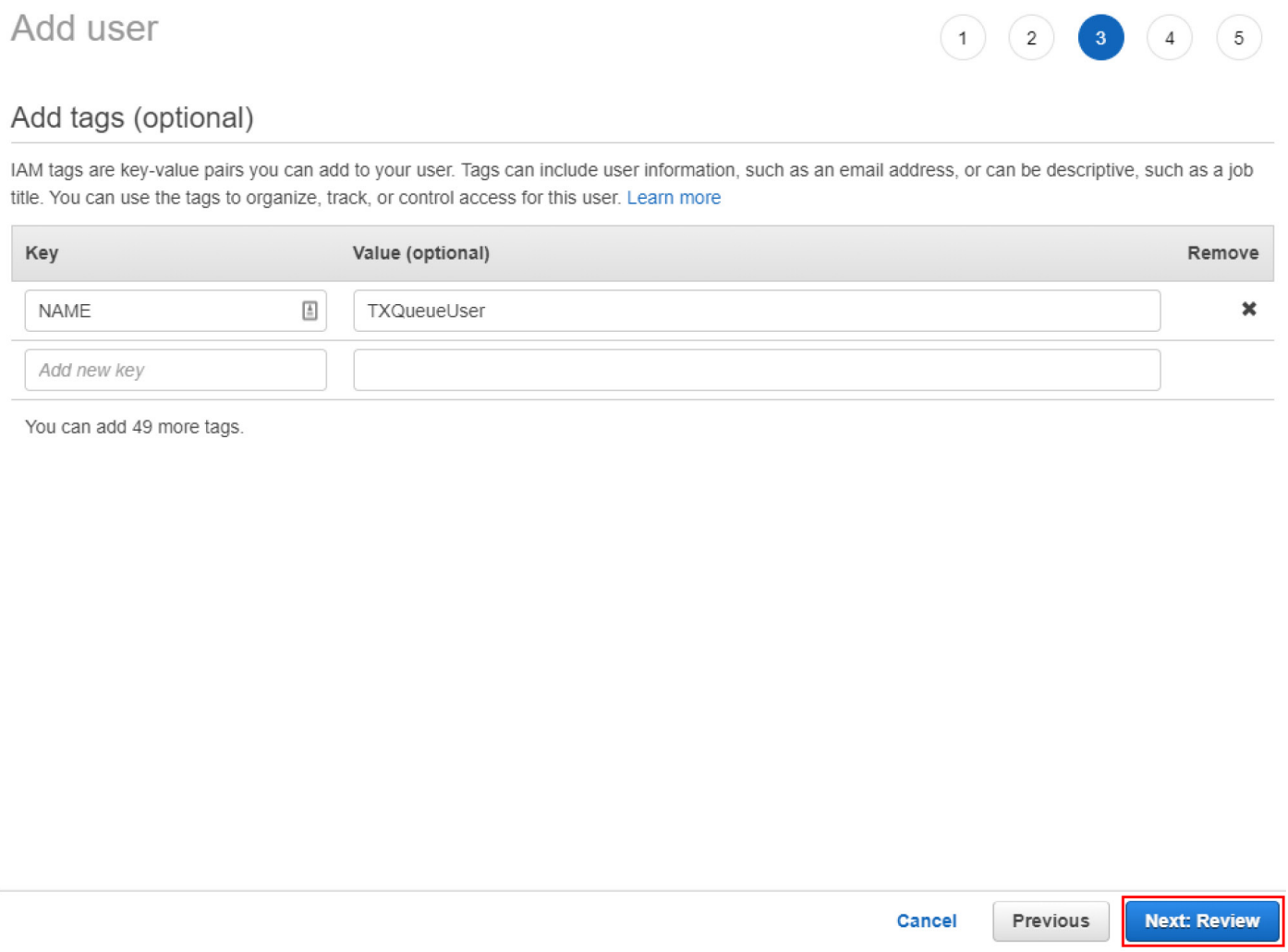
Set Permissions

Choose "Attach existing policies directly", then filter policies by typing policy name created earlier and mark the policy then click on "Next: Tags" button



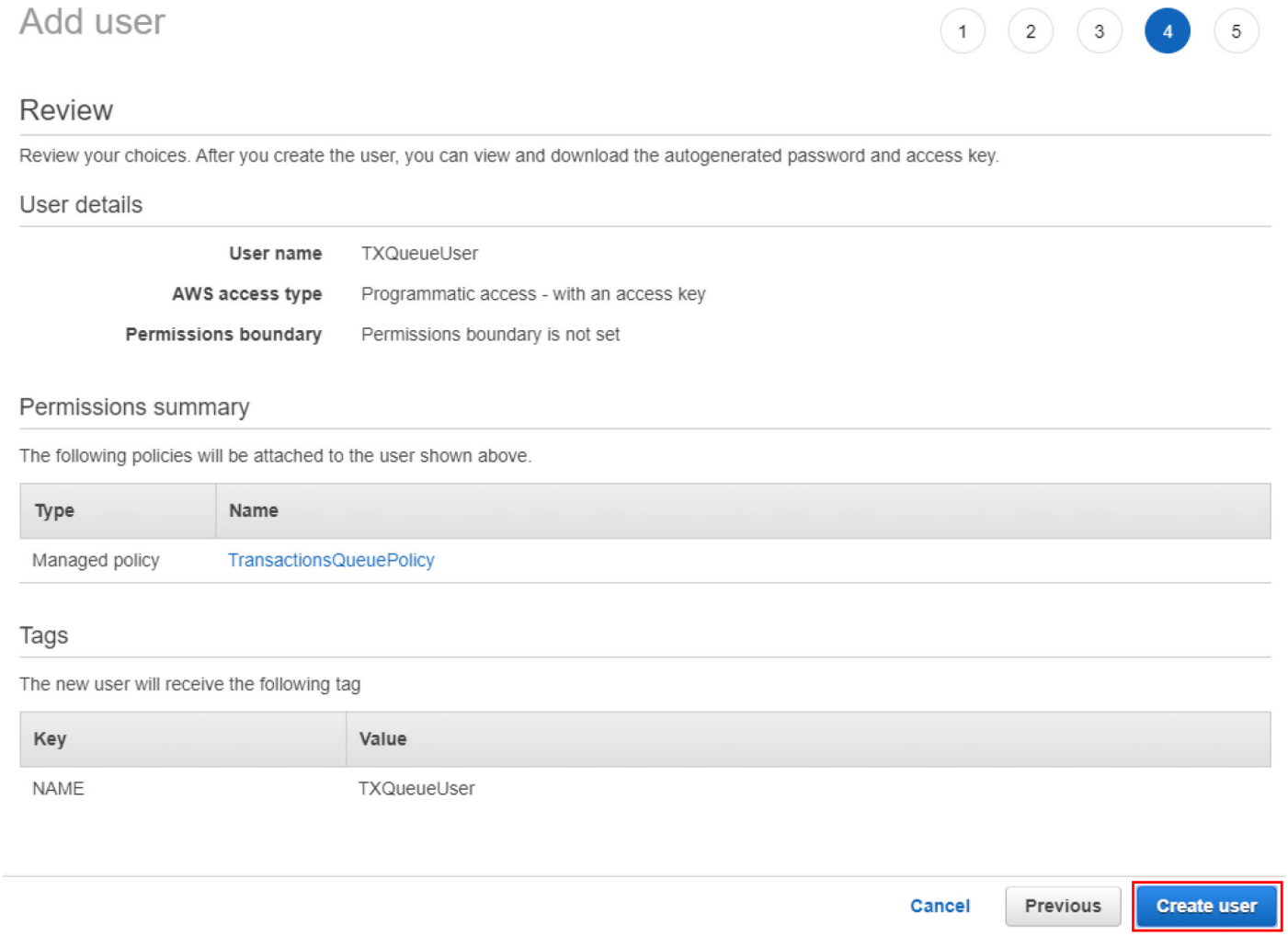
Add tags

AWS tags are used to identify resource pricing calculations – so if you want to see the resource in the AWS expenses report, add the key "NAME" and the value should be the name of the tag you want



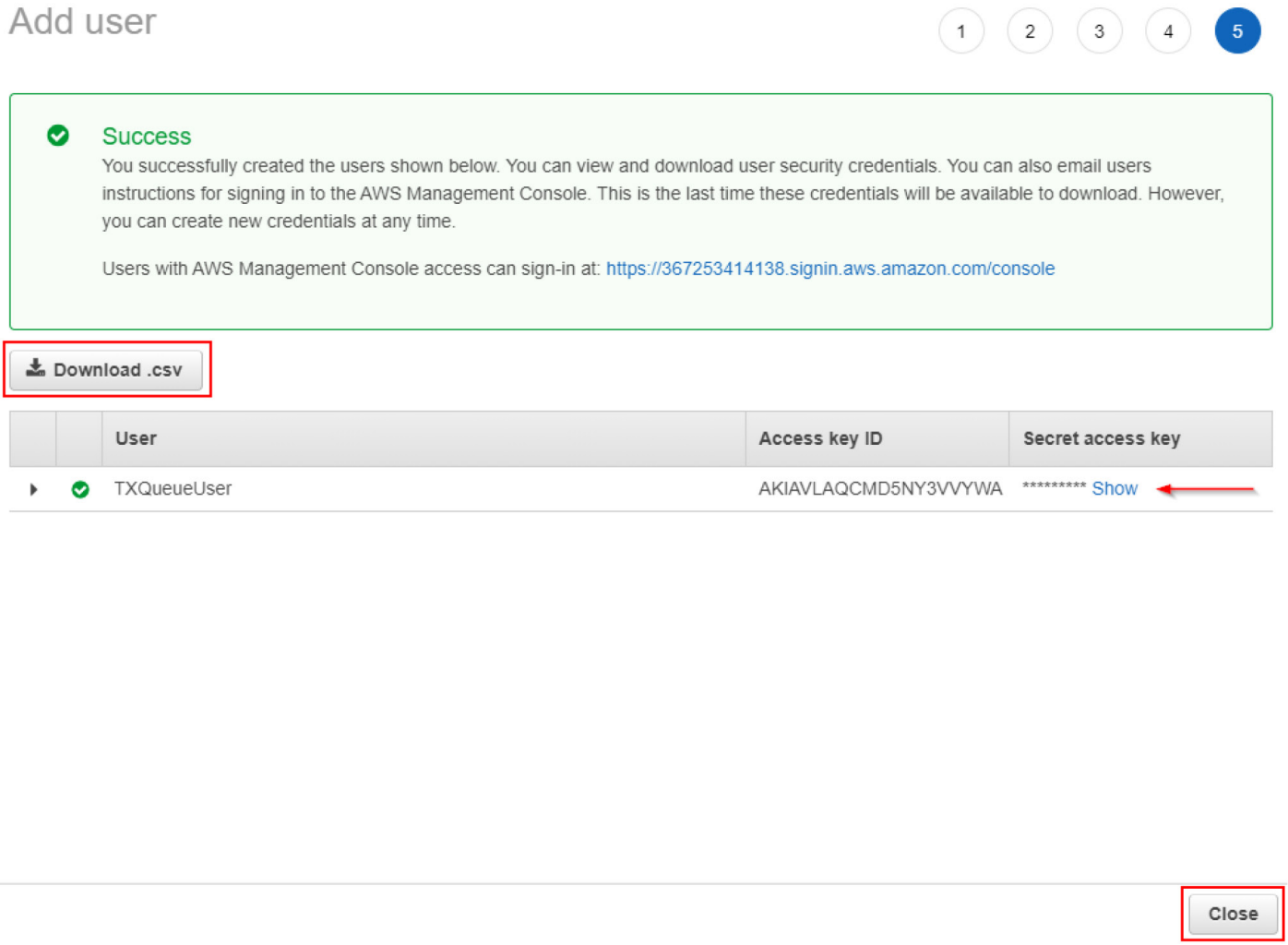
Click "Next: Review" button

Review all the settings to ensure that everything is in order and click on "Create User"



Save Access and Secret Key

Once user is created you will be shown the Access Key and Secret Key , you are able to download these as a CSV file or Click "Show" to view and copy them,
it is recommended to do both



Please Note: This is the only time you will be able to access the Secret Key, please make sure to store it before proceeding to click close

Once you click close, you are done and now should have the required information to access the Queue, The details required to continue the setup in Nayax Back office are **Queue URL**, **Access Key** and **Secret Key**

Nayax Back Office Setup

Transaction Delivery Setup

In order to setup Transaction Delivery to Amazon SQS, you must have access to Nayax back office and Relevant user permissions, Specifically the following roles :

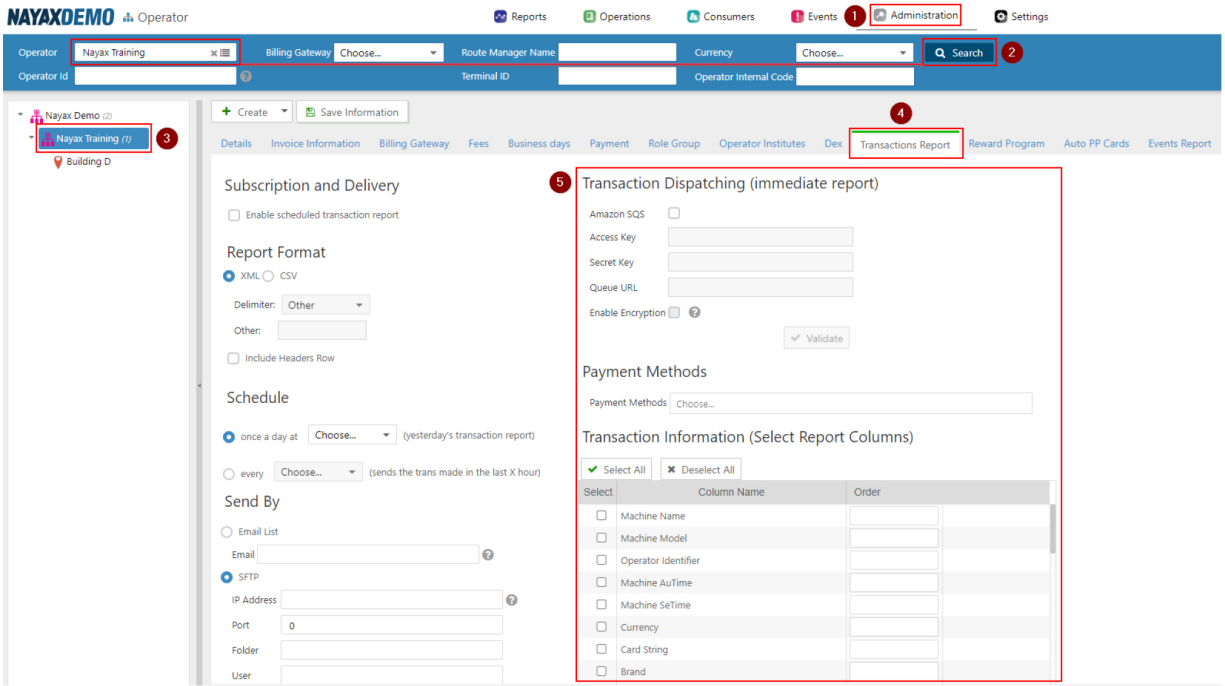
Transaction Dispacther

Transaction Report Subscriber

Prepare the Amazon SQS Queue URL, Access Key and Secret Key obtained when creating the queue

Operator Level Setup

In Nayax Back Office
1. Navigate to Administration > Operator
2. Find the Operator you wish to setup and click search
3. Choose the operator to display all tabs and details
4. Navigate to Transactions Report Tab
5. Setup Transaction Dispatching
    a. Mark Amazon SQS check box and Fill in SQS Credentials
    b. Click "Validate" button to ensure Queue credentials are correct
    c. Enable Encryption – Optional (explained in Setup Encryption Section)
    d. Choose Payment Methods – Only Selected Payment Methods will be sent to Queue
    e. Select Columns of information – see fields description table below
6. Save Information



Example Transaction JSON Message


```
{
 "TransactionId": 3804536984,
 "RemoteStartTransactionId": null,
 "PaymentMethodId": 2,
 "SiteId": 6,
 "MachineTime": "2020-04-20T10:34:05.063",
 "Void": false,
 "MachineId": 54265,
 "Data": {
 "Card String": "0077020491",
 "Settlement Time (Date only)": "2020-04-20",
 "Settlement Time (Time only)": "07:34:15",
 "SeValue": 1.0000,
 "Operator Identifier": "500905",
 "Machine Group": "Snacks",
 "Raw ENI Loyalty Num": null,
 "Machine Name": "Oleg's Simulator 1",
 "Machine AuTime": "2020-04-20T10:34:05.063",
 "Machine SeTime": "2020-04-20T10:34:15.433",
 "Currency": "EUR",
 "Brand": null,
 "CLI": null,
 "Extra Charge": 0.0000,
 "Payment Service(Mobile using Credit Card)": "Prepaid Credit using Prepaid Credit",
 "Payment Method ID (1)": 2,
 "Recognition Method ID (3)": 3,
 "Catalog Number": "",
 "Device Number": "0000000000500902",
 "Actor Hierarchy": "Dually / Nayax Test Bunny / Connect 2020 Rocks / ",
 "Payment Method Description": "Prepaid Credit",
 "Recognition Description": "Prepaid Credit",
 "Card First 4 Digits": "0077",
 "Card Last 4 Digits": null,
 "Card Type": "MIFARE received as Hexadecimal",
 "Transaction ID": 3804536984,
 "Site ID": 6,
 "Authorization Time": "2020-04-20T07:34:05.063",
 "Authorization Value": 6.0000,
 "PayServTransId": "200420073405",
 "sePayServTransId": null,
 "Settlement Time": "2020-04-20T07:34:15.433",
 "Cancel Type": null,
 "Is Revalue Transaction": false,
 "Preselection Status": 0,
 "Is Phone Registration": false,
 "Is Multivend": false,
 "Settlment Failed": null,
 "Sale ID": -1,
 "Sale Value": 0.0000,
 "Updated DT": "2020-04-20T07:34:15.433",
 "Constant Preauthorization Value": null,
 "Is Partial Confirmation": null,
 "Authorization Code": null,
 "Authorization Date and Time": null,
 "Authorization RRN": "200420073405",
 "Event Code": null,
 "Guest Name": null,
 "Token": null,
 "Zip Code": null,
 "Billing Provider ID": null,
 "AVS Only": null,
 "BOD Transaction Key": null,
 "Disable Debit Cards": null,
 "Force Transactions Terminal": null,
 "Use Phone Transaction": null,
 "License ID": null,
 "Merchant ID": null,
 "Billing Site ID": null,
 "Terminal ID": null,
 "User Password": null,
 "With ZIP": null,
 "Use Phone Contactless": false,
 "Use Phone Contact": false,
 "Debit Card Prefix": null,
 "Actor Description": "Connect 2020 Rocks",
 "Institute Description": "NAYAX US",
 "Location Code": 234234,
 "Location Description": "Booth 134",
 "Operator Institute Code": 4,
 "Area Description": "Connect 2020 Rocks",
 "OP Button Code": "Z1",
 "Barcode": "",
 "Cost Price": null,
 "Card Price": null,
 "Prepaid Price": null,
 "Machine Price": 4.0000,
 "Cash Price": null,
 "Default Price": null,
 "Actor Code": 12,
 "Display Card Number": "0077020491",
 "Card Holder Name": "Automatically Created",
 "User Identity": "1",
 "Billing Provider Name": null,
 "Is Offline Transaction": null,
 "Is EMV Transaction": null,
 "Machine AuTime (Date only)": "2020-04-20",
 "Machine AuTime (Time only)": "10:34:05",
 "Machine SeTime (Date only)": "2020-04-20",
 "Machine SeTime (Time only)": "10:34:15",
 "Updated DT (Date only)": "2020-04-20",
 "Updated DT (Time only)": "07:34:15",
 "Customer Type": 1,
 "Actor ID": 32622,
 "Client Id": "",
 "Contract Name": null,
 "Payout Day": null,
 "Contract Id": null,
 "Airport Id": null,
 "Is Refund Card": false,
 "Contract Number": null,
 "Airport Code": null,
 "Payed Value": 1.0000,
 "Consumer ID": 19068140046686389O3,
 "Discount Card ID": 340897206534728,
 "Discount Card Number": "1424243531",
 "Discount Card User Identity": "",
 "Discount Card Physical Type ID": 30000531,
 "Discount Card Activation Date": null,
 "Discount Card Expiration Date": null,
 "Products": [
 {
 "Product(Product Name(MDB Code,PA Code))": "Bounty(0,A1)",
 "Product Group": "Snacks",
 "Product Code in Map": 0,
 "Product PA Code": "A1",
 "Product Volume Type": "600 ml",
 "Product Name": "Bounty",
 "Product VAT Id": null,
 "Product Tax Value": null,
 "Product Tax Code": null,
 "Product Vat Amount": null,
 "Product Net Price": null,
 "Product External Prepaid Price": null,
 "Product Group Code": null,
 "Product Group Sub Code": null,
 "Product Retail Price": null,
 "Product Discount Percentage": 0.00,
 "Product Discount Amount": 0.0000,
 "Product Bruto": 1.0000,
 "Product Catalog Number": ""
 }
 ]
 }
}
```

JSON Message Fields Description

Machine Name	String	Nayax Office Coffee Machine	Machine / POS Name as defined under Operations > Machines > Machine Name
Operator Identifier	String	391453	Machine / POS Number as defined under Operations > Machines > Machine Number
Machine AuTime	Date Time	2020-03-14T11:22:04.930	Machine Authorization Date and Time based on Machine Time Zone
Machine SeTime	Date Time	2020-03-14T11:22:04.930	Machine Settlement Date and Time based on Machine Time Zone
Currency	String	ILS	Transaction Currency Based on Machine Configuration
Card String	String	4925 xxxx xxxx 9940	First 4 and Last 4 digits of the payment card
Brand	String	VISA	Credit Card Brand Name
CLI	String	054-5688038	Consumer Phone Number, applicable when transaction made through Mobile
SeValue	Decimal	1.7000	Settlement Value, the amount charged
Extra Charge	Decimal	0.0000	Extra Charge Convenience fee
Payment Service(Credit Card using Credit Card)	String	Credit Card using Credit Card	Payment and Recognition Method Used in this transaction
Payment Method ID (1)	Integer	1	Payment Method ID (Nayax Internal Identifier)
Recognition Method ID (3)	Integer	3	Recognition Method ID (Nayax Internal Identifier)
Catalog Number	String	CN242343	Product Catalog Number as defined under Administration > Products > Product ID
Product(Product Name(MDB Code,PA Code))	String	Bounty(0,A1)	Sold Product Name with MDB Code and PA Code identifiers as defined in Machine Product Map
Device Number	String	0404040419552293	Nayax Device Serial Number
Actor Hierarchy	String	Dually / Nayax Office / Cafeteria	Actor Hierarchy as Breadcrumbs
Payment Method Description	String	Credit Card	Payment Method Name
Recognition Description	String	Credit Card	Recognition Name
Card First 4 Digits	String	4925	Credit Card First 4 digits
CLI	String	9940	Credit Card Last 4 digits
Card Type	String	MIFARE received as Hexadecimal	Description of Cart Type used in current transaction
Machine Group	String	Office	Machine Group Name as defined on Machine
Product Group	String	Coffee	Sold Item Product Group Name
Transaction ID	Integer	3794004680	Transaction ID (Nayax Internal Identifier)
Site ID	Integer	6	Nayax Servers Site ID
Authorization Time	Date Time	2020-03-24T14:58:34.330	Transaction Authorization Date and Time (GMT)

Field Name	Type	Example	Description
Preselection Status	Integer	0	Transaction Preselection Status, Server value to indicate transaction status on server side (internal use for transaction processing options)
Is Phone Registration	Boolean	false	Bit indicating if transaction is part of Phone Registration (Old Deprecated functionality)
Is Multivend	Boolean	false	Bit indicating if transaction is a Multivend (Multi selection sale/vend)
Settlement Failed	Boolean	false	Bit indicating if transaction settlement has failed
Sale ID	Integer	-1	Sale/Discount ID, holds value if transaction had a discount applied on it
Sale Value	Decimal	0.0000	Sale/Discount Amount
Updated DT	Date Time	2020-03-24T14:58:39.997	Transaction Update Date and Time (GMT)
Constant Preauthorization Value	Decimal	1.7000	Authorization Amount as defined under Operations > Machines > Machine Payment Tab > Constant Preauthorization Field
Is Partial Confirmation	Boolean	false	Bit indicating if a transaction is a partial confirmation (applicable only when Credit Call is the defined billing provider)
Authorization Code	String	52170366	
Authorization Date and Time	Date Time	2020-03-24T14:58:39.997	Authorization Date and Time (GMT)
Authorization RRN	Integer	200420090649	Authorization Retrieval Reference Number – Reference number for transaction.
Event Code	Integer	3	
Guest Name	String		
Token	String		
Zip Code	String		
Billing Provider ID	Integer	14	
AVS Only	Boolean	false	
BOS Transaction Key	String		
Disable Debit Cards	Boolean	false	
Force Transactions Terminal	String		
Use Phone Transaction	Boolean	false	
License ID	Integer		
Merchant ID	String		
Billing Site ID	Integer	0	
Terminal ID	Integer	8500657	
User Name	String		
User Password	Boolean	false	
With ZIP	Boolean	false	
Use Phone Contactless	Boolean	false	
Use Phone Contact	String		
Debit Card Prefix	String		
Actor Description	String	Cafeteria	Actor Hierarchy Entity Name of Machine's Direct Actor
Institute Description	String	NAYAX US	Institute / Customer Name
Location Code	Integer	234234	Institute Location Code
Location Description	String	Booth 134	Institute Location Name
Operator Institute Code	Integer	4	Operator Institute Code as defined under Administration > Operator > Operator Institutes > Code Column
Area Description	String	Some Area Name	

Field Name	Type	Example	Description
OP Button Code	String	Z1	Product OP Button Code as define under Operations > Machines > Machine Product Map > OP Button Code column
Barcode	String	12345	Product Barcode as defined under Administration > Products > Product Barcode Field
Cost Price	Decimal	2.9000	Product Cost Price as defined under Administration > Products > Product Cost Price Field
Card Price	Decimal	6.0000	Bit indicating if transaction settlement has failed
Prepaid Price	Decimal	4.0000	Product Prepaid Card Price as defined under Operations > Machines > Machine Product Map > Credit Card Price Column
Machine Price	Decimal	4.0000	Product Machine Price as displayed under Operations > Machines > Machine Product Map > Machine Price Column
Cash Price	Decimal	6.0000	Product Cash Display Price as defined under Operations > Machines > Machine Product Map > Cash Display Price Column
Default Price	Integer	12	Operator Internal Code Fields as defined under Administration > Operator > Operator Internal Code
Customer Type	Integer	1	
Actor ID	Integer	32622	Actor Hierarchy Entity ID of Machine's Direct Actor
Display Card Number	String	0077020491	Nayax Prepaid Card Display Card Number
Card Holder Name	String	Automatically Created	Nayax Prepaid Card Holder Name
User Identity	String	1	Nayax Prepaid Card Holder User Identity
Billing Provider Name	String	Credit Guard	Billing Provider Name as defined on Machine's Actor Hierarchy
Is Offline Transaction	Boolean	false	Bit indicating if this is an EMV Offline Transaction
Is EMV Transaction	Boolean	false	Bit indicating if this is an EMV Transaction
Machine AuTime (Date only)	Date	2020-04-20	Machine Authorization Date based on Machine Time Zone
Machine AuTime (Time only)	Date	10:34:05	Machine Authorization Time based on Machine Time Zone
Machine SeTime (Date only)	Date	2020-04-20	Machine Settlement Date based on Machine Time Zone
Machine SeTime (Time only)	Time	10:34:15	Machine Settlement Time based on Machine Time Zone
Settlement Time (Date only)	Date	2020-04-20	Machine Settlement Date (GMT)
Settlement Time (Time only)	Time	07:34:15	Machine Settlement Time (GMT)
Updated DT (Date only)	Date	2020-04-20	Transaction Update Date (GMT)
Updated DT (Time only)	Time	07:34:15	Transaction Update Time (GMT)
Raw ENI Loyalty Num	String		
Parsed ENI Loyalty Num	String		
Product PA Code	String	A1	Product PA Code as defined under Operations > Machines > Machine Product Map > PA Code column
Product Volume Type	String	330 ml can	Product Volume Type as define under Administration > Products > Product Volume Type field
Product Name	String		Product Name as define under Administration > Products > Product Name field

Field Name	Type	Example	Description
Product VAT Id	Integer	4006	
Product Tax Value	Decimal	17.00	Product Tax Value as defined under Administration > Products > Product Group VAT Info > Tax Value Field
Product Tax Code	Integer	4321	Product Tax Code as defined under Administration > Products > Product Group VAT Info > Tax Code Field
Product Vat Amount	Decimal	0.2906	Product VAT Amount calculated based on the Tax Value as defined on Product Group
Product Net Price	Decimal	1.7094	Product NET Price calculated by reducing Extra Charge and Product VAT Amounts from Settlement Value
Product External Prepaid Price	Decimal	5.0000	Product External Prepaid Price as defined under Operations > Machines > Machine Product Map > External Prepaid Price Column
Product Group Code	Integer	9876	Product Group Code as defined under Administration > Products > Product Group Info > Code Field
Product Group Sub Code	Integer	6798	Product Group Sub Code as defined under Administration > Products > Product Group Infor > Sub Code Field
Product Retail Price	Decimal	6.0000	Product Retails Price as defined under Operations > Machines > Machine Product Map > Retail Price Column
Product Discount Percentage	Decimal	0.67	
Product Discount Amount	Decimal	4.0000	
Product Bruto	Decimal	2.0000	
Product Catalog Number	String	12345	Product Catalog Number as defined under Administration > Products > Product ID Field
Client Id	String		
Contract Name	String		
Payout Day	Integer	15	
Contract Id	String		
Airport Id	Integer	11892	
Member Type	String		
Is Refund Card	Boolean	false	Bit indicating if this is a Nayax Refund Card
Contract Number	String		
Airport Code	String	TLV	Actor Airport Code as defined under Administration > Operator > Operator Airport Code Field
Payed Value	Decimal	1.0000	Transaction Charge Amount
Consumer ID	Integer	1906814004668638903	Unique Consumer Identifier
Discount Card ID	Integer	340897206534728	Discount Count Card ID (Nayax Internal Unique Identifier)
Discount Card Number	String	1424243531	Discount Card Number – Card Unique Identifier
Discount Card User Identity	String	123456	Discount Card User Identity as defined under Consumers > Cards Management > Discount Card User Identity Field
Discount Card Physical Type ID	String	30000531	Discount Card Physical Type Lut Value ID, example: 30000531 - Contactless Sticker
Discount Card Activation Date	Date Time	2020-04-20T00:00:00.000	Discount Card Activation Date as defined under Consumers > Cards Management > Discount Card Activation Date Field
Discount Card Expiration Date	Date Time	2020-04-30T00:00:00.000	Discount Card Expiration Date as defined under Consumers > Cards Management > Discount Card Expiration Date Field

EVA DTS DEX/DDCMP Audit file Delivery Setup

In order to setup DEX Files Delivery to Amazon SQS, you must have access to Nayax back office and Relevant user permissions, Specifically the following roles:

VMO - Manager

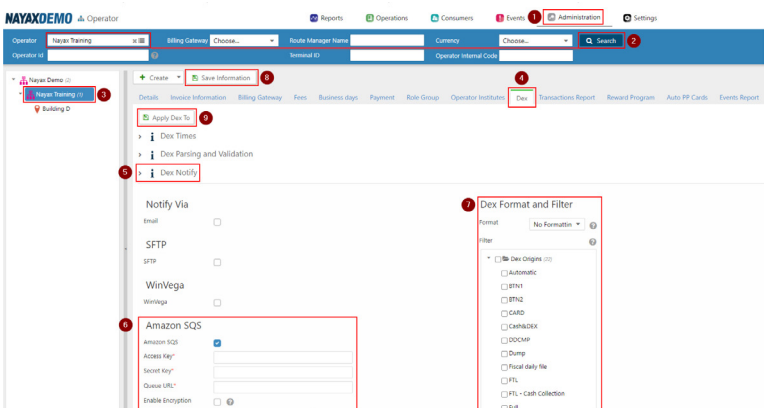
DEX Notify - SQS

Prepare the Amazon SQS Queue URL, Access Key and Secret Key obtained when creating the queue

> Please Note: these setup instructions are with the assumption that DEX Read schedule is already configured

Operator Level Setup

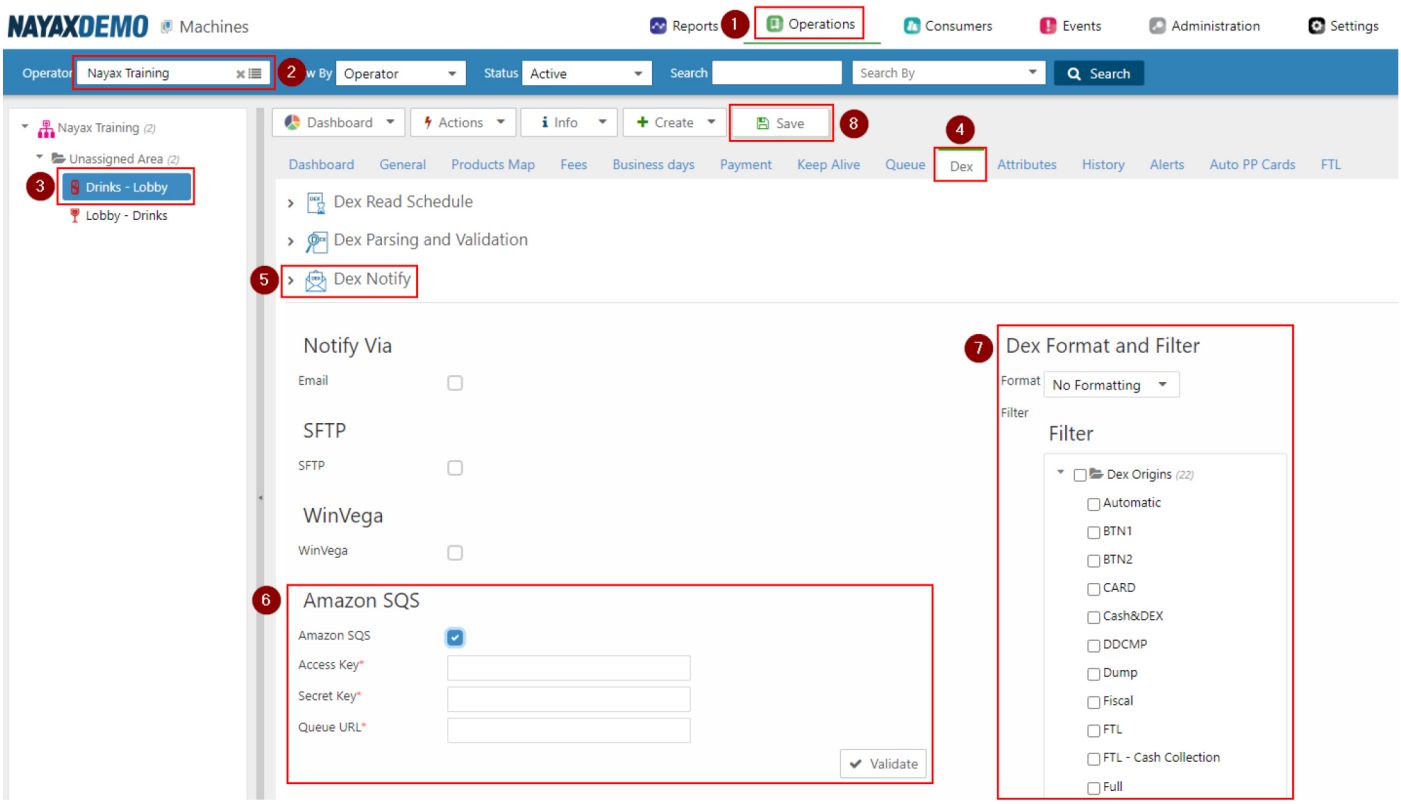
In Nayax Back Office
1. Navigate to Administration > Operator
2. Find the Operator you wish to setup and click search
3. Choose the operator to display all tabs and details
4. Navigate to Dex Tab
5. Click to open Dex Notify Section
6. Setup DEX Transmission to Amazon SQS
a. Mark Amazon SQS check box and Fill in SQS Credentials
b. Click “Validate” button to ensure Queue credentials are correct
c. Enable Encryption – Optional (explained in Setup Encryption Section)
7. Choose Format and Filter
a. Format Should Remain “No Formatting”
b. Choose DEX Read Types – Only Selected Dex Origins will be sent to Queue
8. Save Information
9. DEX Delivery Setup on Operator Level Does NOT initiate delivery of DEX files to Queue, it is used as operator defaults for new machines created under the operator’s hierarchy, you can use the “Apply Dex to” button to propagate the DEX delivery configurations to machines under the operator or perform individual setup on machine level as explained in next article User button



Machine Level Setup

Please Note: these setup instructions are with the assumption that DEX Read schedule is already configured

In Nayax Back Office
1. Navigate to Operations > Machines
2. Find the Operator you wish to setup and click search
3. Choose a machine to display all tabs and details
4. Navigate to Dex Tab
5. Click to open Dex Notify Section
6. Setup DEX Transmission to Amazon SQS
a. Mark Amazon SQS check box and Fill in SQS Credentials
b. Click "Validate" button to ensure Queue credentials are correct
c. Enable Encryption – Optional (explained in Setup Encryption Section)
7. Choose Format and Filter
a. Format Should Remain "No Formatting"
b. Choose DEX Read Types – Only Selected Dex Origins will be sent to Queue
8. Save Information



Example DEX/DDCMP Audit file JSON Message

{
"HWSerial": "534332119000155",
"MachineDate": "20200325 090132_660",
"OperatorIdentifier": "500454",
"Origin": "AUTOMATIC",
"DexData": "DXS*AUTSENSIT3*VA*V0/6*1\r\nST*001*0001\r\nID1*P138140482*AMS SENSIT
III*****6*0\r\nID4*2*0178*1\r\nID5*200325*0358**NA\r\nCB1*P138140482*AMS SENSIT III*3426-
v1.37*2.50\r\nVA1*7037419*11431*0*0*3165272*6786*0*0\r\nCA1*000045003002*ILS-G801-
009*3600\r\nCA2*4110*8*0*0\r\nCA3*0*0*0*0*1950*0*1950*0*0*0\r\nCA4*0*0*100*0\r\nCA6*9823*1014\r\nCA7*0*0*0*0\r\nCA8*0*18800\r\
nCA10*0*0\r\nCA15*0\r
\nBA1*1511000011754*XBA-EUR40000*0120\r\nDA1*3321190000155*DMX - 2011
*0100\r\nDA2*7033309*11423*0*0\r\nDA4*0*0\r\nDA5*0*0*3165272*6786***31038*220\r\nDA9*0*0\r\nPA1*10*550*Kinder Bueno
Whi\r\nPA2*2*1100*0*0\r\nPA1*11*600*dortos green\r\nPA2*241*143500*0*0*156*73500*0*0*0*0\r\nPA1*13*600*Bamba
nugget\r\nPA2*271*161300*0*0*178*94700*0*0*0*0\r\nPA1*15*450*bamba
big\r\nPA2*504*273000*0*0*375*190037*0*0*20*1700\r\nPA1*17*600*Chips\r\nPA2*269*161300*0*0*191*88050*0*0*0*0\r\nPA1*19*600*Chips\r\
nPA2*196*117300*
0*0*145*73350*0*0*0*0\r\nPA1*21*600*milka\r\nPA2*282*168900*0*0*149*66800*0*0*34*3400\r\nPA1*23*350*Bisli
Gril\r\nPA2*453*213350*0*0*315*108300*0*0*5*650\r\nPA1*25*550*Bisli
Gril\r\nPA2*332*174350*0*0*205*81350*0*0*0*0\r\nPA1*27*600*nishnushim\r\nPA2*280*164700*0*0*179*73750*0*0*0*0\r\nPA1*29*600*Abadi\r\
nPA2*254*135500
*0*0*142*54600*0*0*9*900\r\nPA1*31*600*Click BW\r\nPA2*363*233500*0*0*179*89050*0*0*22*2100\r\nPA1*33*700*Clickcariot\r\nPA2*372*24470
0*0*0*165*76500*0*0*0*0\r\nPA1*35*700*Click-BW\r\nPA2*425*289100*0*0*282*163500*0*0*0*0\r\nPA1*37*700*ClickRed\r\nPA2*462*313000*0*0*
237*119900*0*0*0*0\r\nPA1*39*700*crunch
red\r\nPA2*428*292300*0*0*220*118000*0*0*0*0\r\nPA1*40*550*mekupelet\r\nPA2*269*151650*0*0*129*50450*0*0*2*1100\r\nPA1*41*550*Kinder
Bueno
Whi\r\nPA2*494*275350*0*0*301*121000*0*0*0*0\r\nPA1*42*550*Twix\r\nPA2*353*196800*0*0*261*101387*0*0*0*0\r\nPA1*43*550*Pesek
azman\r\nPA2*302*167350*0*0*203*91100*0*0*1*550\r\nPA1*44*600*gardena\r\nPA2*389*232900*0*0*278*87600*0*0*0*0\r\
nPA1*45*600*KitkatChunky\r\nPA2*388*232300*0*0*201*78350*0*0*0*0\r\nPA1*46*600*Mike and
Ike\r\nPA2*370*221400*0*0*233*82950*0*0*0*0\r\nPA1*47*600*Mentos\r\nPA2*285*170700*0*0*164*45450*0*0*0*0\r\nPA1*48*600*must\r\
nPA2*346*205700*0*0
*223*104550*0*0*0*0\r\nPA1*49*500*Must\r\nPA2*272*119765*0*0*200*83950*0*0*12*5388\r\nPA1*50*550*bueno\r\nPA2*432*242150*0*0*216*7635
0*0*0*0*0\r\
nPA1*51*550*Corny Choco Bana\r\nPA2*404*226050*0*0*235*76950*0*0*7*350\r\nPA1*52*600*Corny
Chocolate\r\nPA2*324*184000*0*0*180*77050*0*0*0*0\r\nPA1*53*1350*EXCE\r\nPA2*358*343750*0*0*197*193574*0*0*33*2600\r\
nPA1*54*1200*voltage\r\nPA2*21
4*145700*0*0*111*60150*0*0*3*1500\r\nPA1*55*1350*EXCE\r\nPA2*408*437050*0*0*235*268737*0*0*67*6300\r\nPA1*56*500*Fitness
\r\nPA2*288*165800*0*0*115*43050*0*0*5*4500\r\nPA1*57*500*Fitness \r\nPA2*227*134600*0*0*102*28100*0*0*0*0\r\nPA1*58*500*Fitness
\r\nPA2*143*82500*0*0*84*23137*0*0*0*0\r\nPA1*59*100*Embedded
Testing****1\r\nPA2*31*15004*0*0\r\nEA1*EGS*200205*0104***58\r\nEA1*EGT*200205*0112***63\r\nEA1*EGS*200205*0412***57\r\
nEA1*EGT*200205*0418***66\r\
nEA1*EGS*200205*0454***56\r\nEA1*EGT*200205*0455***59\r\nEA1*EGS*200205*0456***62\r\nEA1*EGT*200205*0457***64\r\
nEA1*EGS*200205*0705***57\r\nEA1
*EGT*200205*0708***63\r\nEA7*0*151\r\nEA3*6923*200325*0358*1234567890*200325*0307*1234567890**9823*6923\r\
nEA4*151213*0630*1234567890\r\nSD1****
*AUTO*AUTO\r\nMA5*TEMP*59*15\r\nG85*3CE2\r\nSE*110*0001\r\nDXE*1*1\r\n"
}

JSON Message Fields Description

Field Name	Type	Example	Description
HWSerial	String	404044218531269	Nayax Device Serial Number
MachineDate	String	20200325 090132_660	EVA DTS Audit File Readout Date and Time based on Machine Time Zone
OperatorIdentifier	String	500454	Machine / POS Number as defined under Operations > Machines > Machine Number
Origin	String	AUTOMATIC	EVA DTS Audit File read initiated by: Automatic – server request, Scheduled – local device schedule, Button Press – User Initiated etc.
DexData	String	DXS*AUTSENSIT3*VA*V0/6*1\r\n…	EVA DTS Audit File raw data

Events Delivery Setup

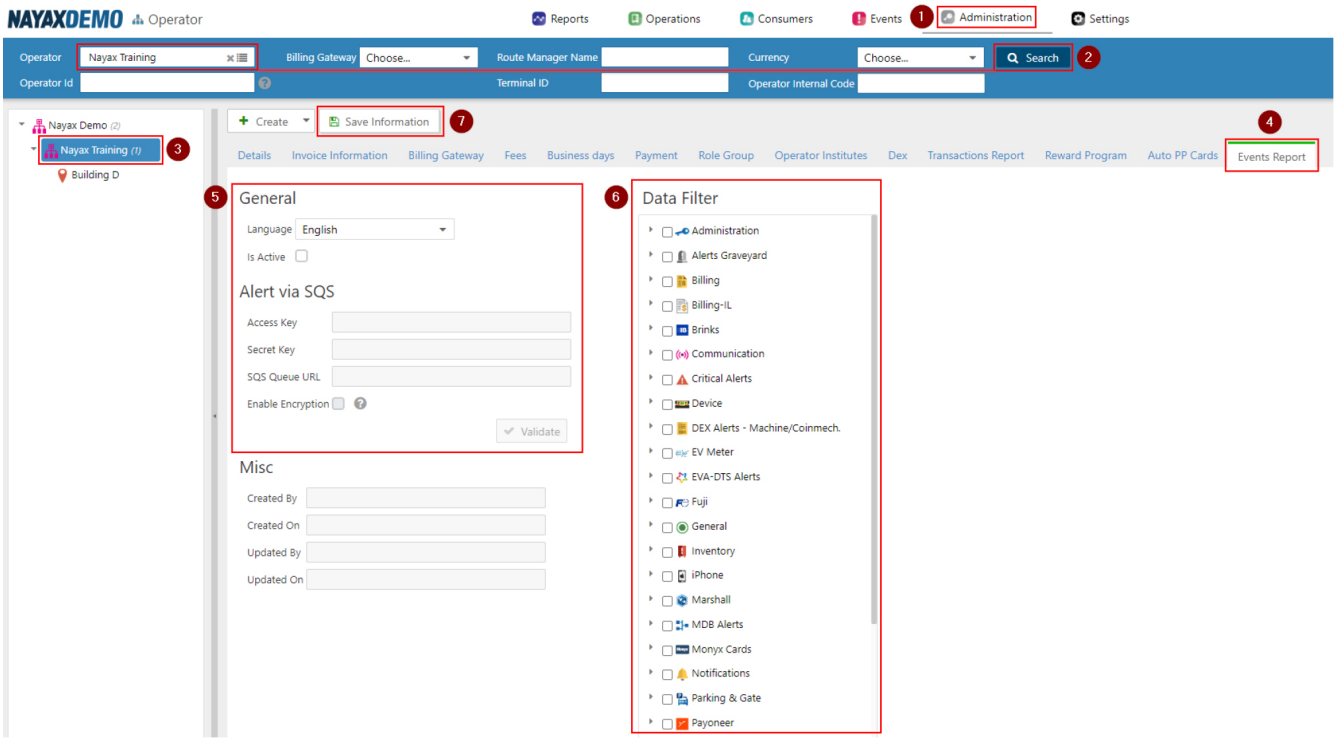
In order to setup Events and Alerts Delivery to Amazon SQS, you must have access to Nayax back office and Relevant user permissions, Specifically the following roles:

VMO - Manager

Prepare the Amazon SQS Queue URL, Access Key and Secret Key obtained when creating the queue

Operator Level Setup

In Nayax Back Office
1. Navigate to Administration > Operator
2. Find the Operator you wish to setup and click search
3. Choose the operator to display all tabs and details
4. Navigate to Events Report Tab
5. Setup DEX Transmission to Amazon SQS
a. Mark is Active check box and Fill in SQS Credentials
b. Click "Validate" button to ensure Queue credentials are correct
c. Enable Encryption – Optional (explained in Setup Encryption Section)
6. Data Filter - Choose Event Types for Delivery
a. Format Should Remain "No Formatting"
b. Choose DEX Read Types – Only Selected Dex Origins will be sent to Queue
7. Save Information



Example Event/Alert JSON Message

{
"distributorId": "27594",
"distributorName": "Nayax Demo",
"operatorId": "2000144542",
"operatorName": "Daniel_OP",
"actorId": "2000144542",
"actorName": "Daniel_OP",
"machineId": "697356467",
"machineName": "EX PP Machine",
"machineNumber": null,
"machineSerial": null,
"deviceId": "25795395",
"deviceSerial": "0404041919567757",
"eventSourceId": "6",
"eventSourceName": "Device",
"eventCode": "1",
"eventName": "Power up",
"eventData": "Device Serial #:0404041919567757\nDevice FW Version:4.0.0.5-rc3 - Jan 9 2019\nDevice Hardware:4.13\nVPOS Serial #:\nVPOS FW Version:2009-rc07\nMDB Max Response Time:7\nDevice MDB Level:1\nDecimal Place:2\nManufacturer Code:4805701\nMachine MDB Level:1\nModem IMEI:357036091438914\nModem Model(HW Revision):UE910-EUD\nModem FW Version:12.00.414\nModem Type:3G\nSIM Card Serial #:89462008002001388010\nSIM Card Mobile Number:NO CLIn\nMobile Operator Name:42501\nModem CDMA SN (MEID):\nLast Power Down time:23/03/2020
14:27:30",
"eventDateTimeGMT": "2020-03-23T14:11:35.980",
"eventDateTimeVMC": "2020-03-23T17:11:35.980",
"enc_ver": "1",
"enc_key": "63562b7245cb478cb38a509ec764e012"
}

JSON Message Fields Description

Field Name	Type	Example	Description
distributorId	String	27594	Distributor Actor ID (Nayax Internal ID)
distributorName	String	Nayax Demo	Distributer Actor Name
operatorId	String	2000144542	Operator Actor ID (Nayax Internal ID
operatorName	String	Daniel_OP	Operator Name
actorId	String	2000144542	Actor Hierarchy Entity ID of Machine's Direct Actor
actorName	String	Daniel_OP	Actor Hierarchy Entity Name of Machine's Direct Actor
machineId	String	697356467	Machine ID (Nayax Internal ID)
machineName	String	EX PP Machine	Machine / POS Name as defined under Operations > Machines > Machine Name
machineNumber	String	3423423	Machine / POS Number as defined under Operations > Machines > Machine Number
machineSerial	String	23443452	Machine / POS Number as defined under Operations > Machines > Machine Serial Number
deviceId	String	25795395	Device ID (Nayax Internal ID)
deviceSerial	String	0404041919567757	Nayax Device Serial Number
eventSourceId	String	6	Event Source ID (Nayax Internal ID)
eventSourceName	String	Device	Event Source Name
eventCode	String	1	Event Code
eventName	String	Power up	Event Name
eventData	String	Device Serial #:0404041919567757\n...	Event Data
eventDateTimeGMT	String	2020-03-23T14:11:35.980	Event Date and Time (GMT)
eventDateTimeVMC	String	2020-03-23T17:11:35.980	Event Date and Time based on Machine Time Zone
enc_ver	String	1	Encryption Key Version
enc_key	String	63562b7245cb478cb38a509ec764e012	Encryption Key

Setup Encryption on Amazon SQS Delivery

In order to setup Encryption on Delivery to Amazon SQS, you must have access to Nayax back office and Lynx API with relevant user permissions, Specifically the following roles:

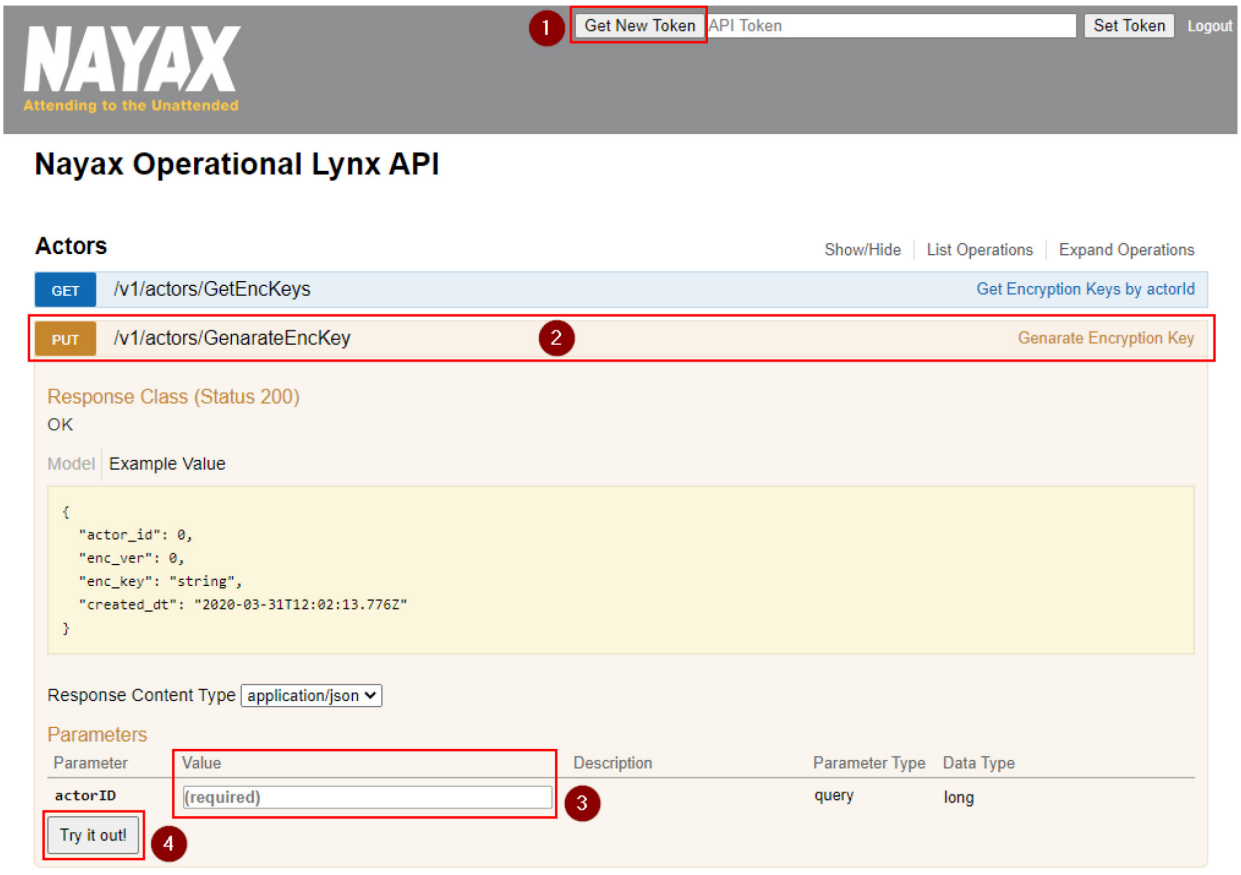
Lynx - SQS Encryption

To enable the encryption on the messages sent Amazon SQS you should mark the "Enable Encryption" check box in the relevant configuration section as described in the previous sections.

Open a web browser and Navigate to the following URL:https://lynx.nayax.com/operational/signin
There you will be prompted to Login, use your Nayax Back office credentials

Generate Encryption Keys

For next step please prepare Actor ID (Operator ID) which can be found under Administration > Operator >
Details Tab
1. Click on Generate New Token Button in the top right corner of the screen
2. Click on "PUT /v1/actors/GenerateEncKey" row to reveal more details
3. Fill in actorID Parameter value
4. Click "Try it out!" button



Review all the settings to ensure that everything is in order and click on "Create User"

Field Name	Description
actor_id	Auto generate unique id, represents the hierarchical entity in the hierarchy tree
enc_ver	Numerator identifying Encryption Key Version
enc_key	Encryption Key, Alphanumeric GUID used to encrypt Messages delivered to Amazon SQS
created_dt	Encryption Key Creation Date and Time

Example Response

Curl

```
curl -X PUT --header 'Accept: application/json' --header 'Authorization: Bearer OBQjt0TtJbdDG5ViLhO7G5_jJKb6JaoGX7X9pTO-WxToeuSUhA
```

Request URL

```
https://lynx.nayax.com/Operational/v1/actors/GenarateEncKey?actorID=39517
```

Response Body


```
{
  "actor_id": 39517,
  "enc_ver": 5,
  "enc_key": "fe64b0da60264ed295403966514b8b33",
  "created_dt": "2020-03-31T12:19:09.4966834Z"
}
```


Response Code

```
200
```

Response Headers


```
{
  "access-control-allow-headers": "Content-Type",
  "access-control-allow-methods": "GET, POST, PUT, DELETE, OPTIONS",
  "access-control-allow-origin": "*",
  "cache-control": "no-cache",
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Tue, 31 Mar 2020 12:19:09 GMT",
  "expires": "-1",
  "pragma": "no-cache",
  "server": "Cant touch this ta na na na",
  "strict-transport-security": "max-age=480; includeSubDomains",
  "transfer-encoding": "chunked",
  "vary": "Accept-Encoding",
  "x-aspnet-version": "4.0.30319"
}
```


If the "Enable Encryption" checkbox is marked for any of the SQS message delivery setup in Nayax Back office, messages sent to Queue will start to be encrypted with the given encryption key, This key can be refreshed by generating a new one as described in the steps above and a new key will be provided with enc_ver value incremented to indicate a Key Version Number.

List Encryption Keys

All previously generated keys can be obtained through the GET Encryption Keys option
1. Click on "GET /v1/actors/GetEncKeys" row to reveal more details
2. Fill in actorID Parameter value
3. Click "Try it out!" button

Example Response: