

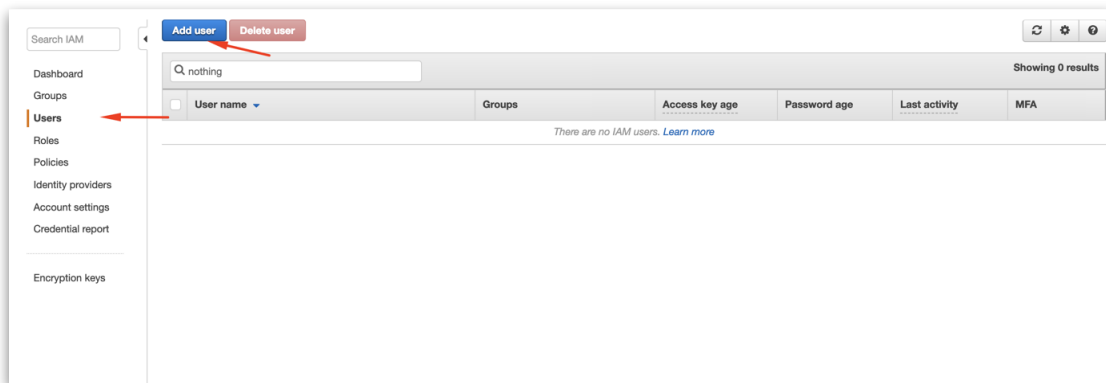
Integrating AVANAN with AWS S3 for Splunk logs – Part Two

Step-1:

- Go to AWS IAM: <https://console.aws.amazon.com/iam/home#/home>
- Note: The AWS configuration is similar to part 1, but the Splunk AWS app requires way more permissions than Avanan. Therefore, to limit Avanan's access to your S3 environment, you'll need to create a new user, group, policy, and role to use on Splunk.

Step-2:

Click on Users > Add User



Step-3:

- Select a name and enable “Programmatic access”, click “Next: Permissions”

The screenshot shows the 'Add user' form in the AWS IAM console. At the top, there are five numbered steps: 1 (selected), 2, 3, 4, and 5. The form is titled 'Add user' and has a sub-header 'Set user details'. Below this, a message states: 'You can add multiple users at once with the same access type and permissions. [Learn more](#)'. The 'User name*' field contains the text 'splunk-s3-user'. Below the field is a link that says 'Add another user' with a plus icon. The next section is 'Select AWS access type', with a message: 'Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)'. Under 'Access type*', there are two options: 'Programmatic access' (selected with a checked checkbox) and 'AWS Management Console access' (unchecked). The 'Programmatic access' option has a description: 'Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.' The 'AWS Management Console access' option has a description: 'Enables a **password** that allows users to sign-in to the AWS Management Console.' At the bottom of the form, there is a '* Required' label, a 'Cancel' button, and a 'Next: Permissions' button.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required Cancel Next: Permissions

Step-4

- Click on “Create group” (or the right group if already created)

The screenshot shows the 'Create group' form in the AWS IAM console. At the top, there is a title bar 'Create group' with a close button. Below the title bar, a message states: 'Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)'. The 'Group name' field is empty, and a red arrow points to it. Below the field are two buttons: 'Create policy' and 'Refresh'. Below these buttons is a 'Filter policies' section with a dropdown menu and a search bar containing the text 'nothing'. Below the search bar is a table with the following columns: 'Policy name', 'Type', 'Used as', and 'Description'. The table is currently empty, and the text 'No results' is displayed below it. At the bottom of the form, there are two buttons: 'Cancel' and 'Create group'.

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Create policy Refresh

Filter policies Showing 0 results

Policy name	Type	Used as	Description
-------------	------	---------	-------------

No results

Cancel Create group

Step-5:

- On the new tab, click on JSON and copy this over:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:GetQueueAttributes",
        "sqs:ListQueues",
        "sqs:ReceiveMessage",
        "sqs:GetQueueUrl",
        "sqs:SendMessage",
        "sqs:DeleteMessage",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:GetBucketTagging",
        "s3:GetAccelerateConfiguration",
        "s3:GetBucketLogging",
        "s3:GetLifecycleConfiguration",
        "s3:GetBucketCORS",
        "config:DeliverConfigSnapshot",
        "config:DescribeConfigRules",
        "config:DescribeConfigRuleEvaluationStatus",
        "config:GetComplianceDetailsByConfigRule",
        "config:GetComplianceSummaryByConfigRule",
        "iam:GetUser",
        "iam:ListUsers",
        "iam:GetAccountPasswordPolicy",
        "iam:ListAccessKeys",
```

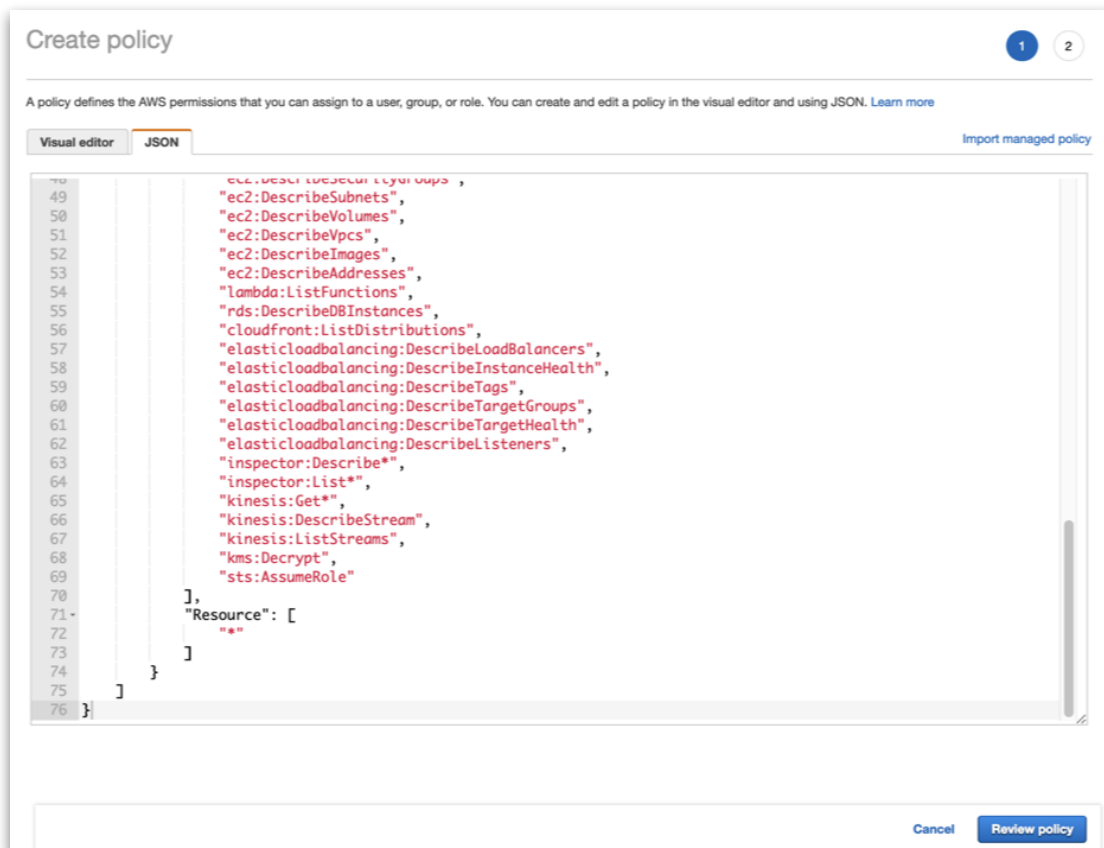
"iam:GetAccessKeyLastUsed",
"autoscaling:Describe*",
"cloudwatch:Describe*",
"cloudwatch:Get*",
"cloudwatch:List*",
"sns:Get*",
"sns:List*",
"sns:Publish",
"logs:DescribeLogGroups",
"logs:DescribeLogStreams",
"logs:GetLogEvents",
"ec2:DescribeInstances",
"ec2:DescribeReservedInstances",
"ec2:DescribeSnapshots",
"ec2:DescribeRegions",
"ec2:DescribeKeyPairs",
"ec2:DescribeNetworkAcls",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVolumes",
"ec2:DescribeVpcs",
"ec2:DescribeImages",
"ec2:DescribeAddresses",
"lambda:ListFunctions",
"rds:DescribeDBInstances",
"cloudfront:ListDistributions",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeInstanceHealth",
"elasticloadbalancing:DescribeTags",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:DescribeListeners",
"inspector:Describe*",

```

    "inspector:List*",
    "kinesis:Get*",
    "kinesis:DescribeStream",
    "kinesis:ListStreams",
    "kms:Decrypt",
    "sts:AssumeRole"
  ],
  "Resource": [
    "*"
  ]
}
]
}

```

- For example:



Step-6:

- Click on Review Policy.
- On the next screen, select a policy name and click on Create Policy:

The screenshot shows the 'Create policy' page in the AWS IAM console, specifically the 'Review policy' step. The page has a header 'Create policy' with step indicators 1 and 2. The 'Review policy' section includes a 'Name' field with the value 'splunk-s3-policy' and a 'Description' field. Below these is a 'Summary' section with a search filter and a table of services and their permissions.

Review policy

Name*
Use alphanumeric and '+=,@_-' characters. Maximum 128 characters.

Description
Maximum 1000 characters. Use alphanumeric and '+=,@_-' characters.

Summary

Service	Access level	Resource	Request condition
Allow (18 of 171 services) Show remaining 153			
CloudFront	Limited: List	All resources	None
CloudWatch	Full: List, Read	All resources	None
CloudWatch Logs	Limited: List, Read	All resources	None
Config	Limited: List, Read	All resources	None
EC2	Limited: List	All resources	None
EC2 Auto Scaling	Full: List, Read	All resources	None
ELB	Full: List Limited: Read	All resources	None
ELB v2	Limited: Read	All resources	None
IAM	Limited: List, Read	All resources	None
Inspector	Full: List Limited: Read	All resources	None
Kinesis	Limited: List, Read	All resources	None
KMS	Limited: Write	All resources	None

* Required

[Cancel](#) [Previous](#) [Create policy](#)

Step-7:

- After the policy is created, go back to the previous tab and click Refresh.
- Select the policy you just created, give the group a name and click on Create group.

The screenshot shows the 'Create group' page in the AWS IAM console. It includes a 'Group name' field with the value 'splunk-s3-group', a 'Create policy' button, and a 'Refresh' button. Below these is a 'Filter policies' section with a search filter and a table of policies.

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

[Create policy](#) [Refresh](#)

Filter policies Showing 1 result

	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	splunk-s3-policy	Customer managed	None	

[Cancel](#) [Create group](#)

Step-8:

- Back to the “Add user” screen, confirm that the group you just created is selected and click on “Next: Tags”

Add user

12345

Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

Create group Refresh

Q splunk-s3

Showing 1 result

Group	Attached policies
<input checked="" type="checkbox"/> splunk-s3-group	splunk-s3-policy

Set permissions boundary

Cancel

Previous

Next: Tags

Step-9:

- Add the necessary Tags (in accordance with your environment directives) and click on “Next: Review”
- Confirm all the configurations and click on “Create user”

Add user 1 2 3 **4** 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	splunk-s3-user
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	splunk-s3-group

Tags

The new user will receive the following tag

Key	Value
Name	splunk-s3-user

[Cancel](#) [Previous](#) [Create user](#)

- Download the CSV or copy the Access Key and Secret access key somewhere safe.
- It will be used to configure Splunk. This information won't be available again.
- Close.

Step-10:

- Click on Roles and on “Create role”
- Select Another AWS Account
- Insert the 12 digit number of your account and click on “Next: Permissions”

The screenshot shows the 'Create role' wizard in the AWS IAM console. The title is 'Create role' with a progress indicator showing steps 1, 2, 3, and 4. Step 1 is 'Select type of trusted entity'. Below the title, there are four options: 'AWS service' (EC2, Lambda and others), 'Another AWS account' (Belonging to you or 3rd party), 'Web identity' (Cognito or any OpenID provider), and 'SAML 2.0 federation' (Your corporate directory). The 'Another AWS account' option is selected. Below these options, there is a text input field for 'Account ID*' with the value '731485868276'. Below the input field, there are two options: 'Require external ID (Best practice when a third party will assume this role)' and 'Require MFA'. At the bottom right, there are 'Cancel' and 'Next: Permissions' buttons.

Create role

1 2 3 4

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID* 731485868276 ⓘ

Options ☐ Require external ID (Best practice when a third party will assume this role)
☐ Require MFA ⓘ

* Required

Cancel Next: Permissions

- Note: to find the 12 digit number, open the user on another screen:

The screenshot shows the 'Summary' page for a user in the AWS IAM console. The user is 'splunk-avanan-user'. The 'User ARN' is 'arn:aws:iam::731485868276:user/splunk-avanan-user'. The 'Path' is '/'. The 'Creation time' is '2019-03-04 14:24 EST'. Below the summary, there are tabs for 'Permissions', 'Groups (1)', 'Tags (1)', 'Security credentials', and 'Access Advisor'. The 'Permissions' tab is selected. It shows 'Permissions policies (1 policy applied)'. There is a button 'Add permissions' and a link 'Add inline policy'. Below this, there is a table with columns 'Policy name' and 'Policy type'. The table shows one policy: 'splunk-avanan-policy' with type 'Managed policy from group splunk-avanan-group'. At the bottom, there is a section for 'Permissions boundary (not set)'.

Users > splunk-avanan-user

Summary

Delete user ⓘ

User ARN arn:aws:iam::731485868276:user/splunk-avanan-user ⓘ

Path /

Creation time 2019-03-04 14:24 EST

Permissions Groups (1) Tags (1) Security credentials Access Advisor

Permissions policies (1 policy applied)

Add permissions Add inline policy

Policy name	Policy type
splunk-avanan-policy	Managed policy from group splunk-avanan-group

Permissions boundary (not set)

Step-11:

- Select the policy you created, click on Next: Tags

Create role

1234

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy↻

Filter policies ▼

🔍 splunk-s3-policy

Showing 1 result

	Policy name ▼	Used as	Description
<input checked="" type="checkbox"/>	▶ splunk-s3-policy	Permissions policy (1)	

▶ Set permissions boundary

* Required

CancelPreviousNext: Tags

Step-12:

- Add the necessary Tags (in accordance with your environment directives) and click on “Next: Review”
- Select a role name and click on Create Role

Create role

1234

Review

Provide the required information below and review this role before you create it.

Role name*

splunk-s3-role

Use alphanumeric and '+=,@-_' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+=,@-_' characters.

Trusted entities

The account 731485868276

Policies

splunk-s3-policy [↗](#)

Permissions boundary

Permissions boundary is not set

The new role will receive the following tag

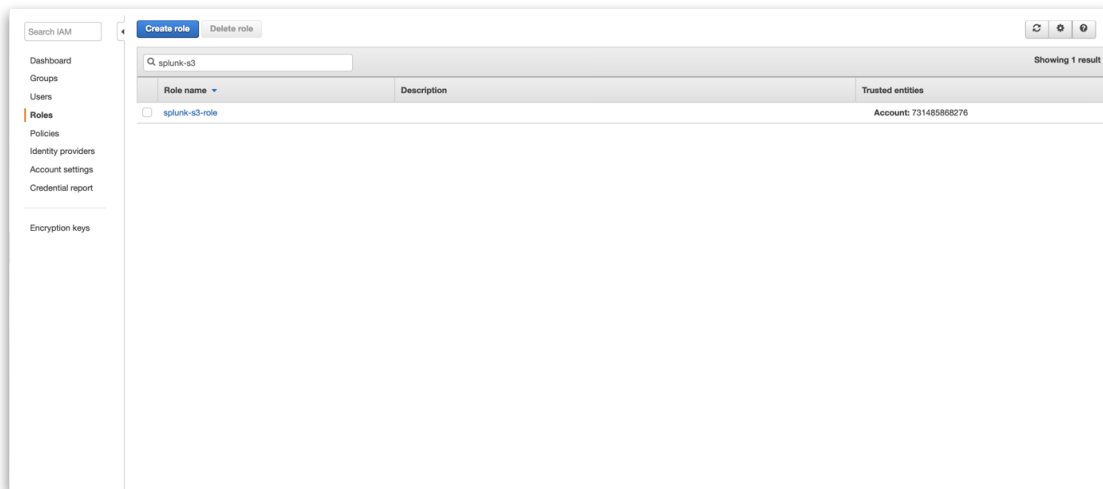
Key	Value
Name	splunk-s3-role

* Required

CancelPreviousCreate role

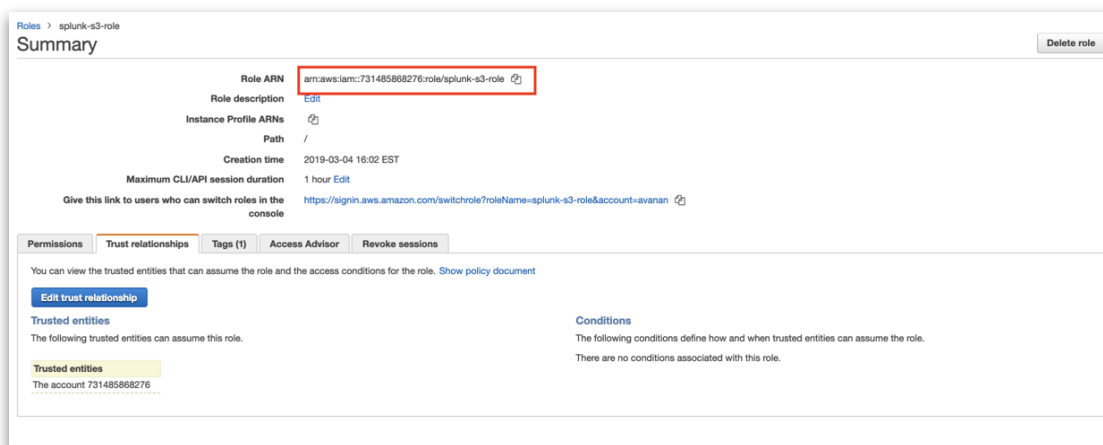
Step-13:

- Search for the role you just created, click on its name.



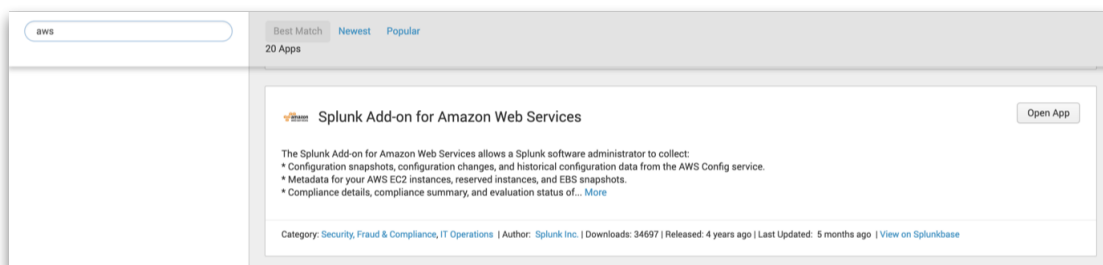
Step-14:

- Copy the role ARN



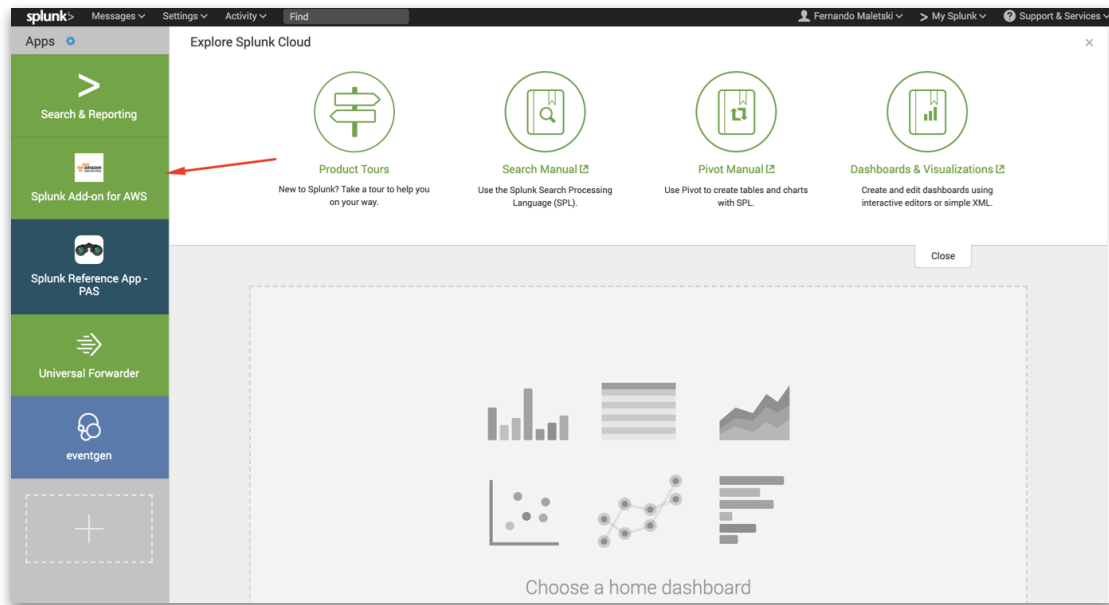
Step-15:

- Now on Splunk, install the AWS add-on if you haven't already, by clicking on the big + and searching for it:



Step-16:

- Click on Open App after it is installed, or on Splunk Add-on for AWS on the main screen



Step-17:

- Click on Configuration > Account > Add and configure it with the Key ID and Secret Key from the CSV you served when the user was created, click on Add:

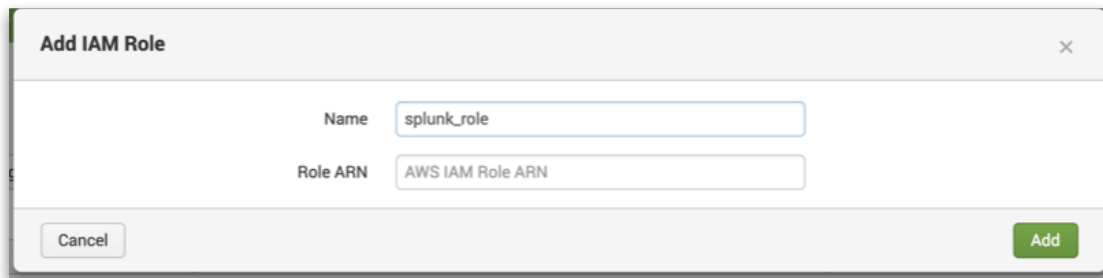
The 'Add Account' dialog box is shown. It has a title bar with 'Add Account' and a close button. The form contains the following fields:

- Name:** A text input field with the value 'splunk_user'.
- Key ID:** A text input field with the value 'AWS account key id'.
- Secret Key:** A password input field with masked characters '*****'.
- Region Category:** A dropdown menu with 'Global' selected.

At the bottom, there are two buttons: 'Cancel' on the left and 'Add' on the right.

Step-18:

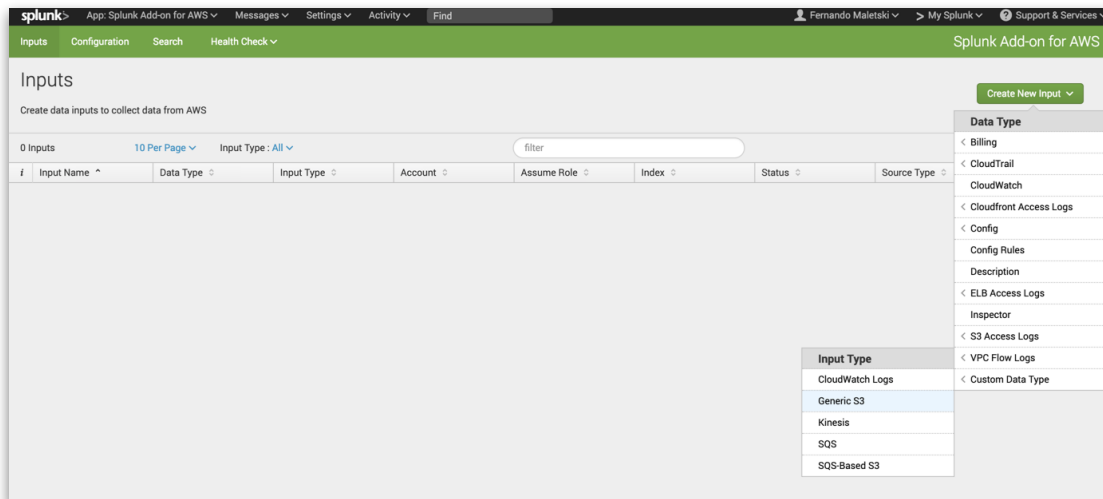
- Click on IAM Role > Add and configure it with the Role ARN you copied previously.



The 'Add IAM Role' dialog box is shown. It has a title bar with a close button. Inside, there are two input fields: 'Name' with the value 'splunk_role' and 'Role ARN' with the value 'AWS IAM Role ARN'. At the bottom, there are 'Cancel' and 'Add' buttons.

Step-19:

- Click on Inputs > Create New Input > Custom Data Type > Generic S3



The screenshot shows the 'Inputs' page of the 'Splunk Add-on for AWS' interface. The page has a header with navigation tabs: 'Inputs', 'Configuration', 'Search', and 'Health Check'. Below the header, there's a 'Create data inputs to collect data from AWS' section. A table with columns 'Input Name', 'Data Type', 'Input Type', 'Account', 'Assume Role', 'Index', 'Status', and 'Source Type' is shown, currently empty. On the right side, there's a 'Create New Input' button and a 'Data Type' dropdown menu. The 'Data Type' menu is open, showing options like 'Billing', 'CloudTrail', 'CloudWatch', 'Cloudfront Access Logs', 'Config', 'Config Rules', 'Description', 'ELB Access Logs', 'Inspector', 'S3 Access Logs', 'VPC Flow Logs', and 'Custom Data Type'. The 'Input Type' dropdown is also open, showing options like 'CloudWatch Logs', 'Generic S3', 'Kinesis', 'SQS', and 'SQS-Based S3'. The 'Generic S3' option is highlighted.

Step-20:

- Select a name for the Input, the AWS Account and the Assume Role you configured above, the S3 Bucket Avanan is uploading the logs, a start datetime (ideally, a few minutes before you enabled Splunk on Avanan as part of Part 1).
- Click on the arrow to show the Advanced Settings and set the Polling Interval to 300 s (5 minutes) as Avanan will upload the logs every 5 minutes.
- Note: We also upload the logs every time they reach 5 MB before 5 minutes (unlikely).
- Click Save.

The screenshot shows the 'Generic S3' configuration page in the Splunk interface. The page has a green header with navigation tabs: 'Inputs', 'Configuration', 'Search', and 'Health Check'. The title 'Generic S3' is at the top left, with a link 'Inputs > Create New Input'. The main content area is titled 'AWS Input Configuration' and contains several sections:

- AWS Input Configuration:** Includes fields for 'Name' (set to 'avanan_json_data'), 'AWS Account' (set to 'splunk_user'), 'Assume Role' (set to 'splunk_role'), 'S3 Bucket' (set to 'avanan-splunk-test'), and 'S3 Key Prefix' (set to 'optional').
- Splunk-related Configuration:** Includes 'Start Date/Time' (set to '2019-02-20T21:14:27Z'), 'End Date/Time' (set to 'e.g., 2000-01-01T00:00:00Z (optional)'), 'Source Type' (set to 'aws:s3'), and 'Index' (set to 'default').
- Advanced Settings:** Includes 'Blacklist' (set to 'optional'), 'Whitelist' (set to 'optional'), and 'Polling Interval (in seconds)' (set to '300').

At the bottom right, there are 'Cancel' and 'Save' buttons.

- And done. The Splunk logs are now being read from the S3 bucket where Avanan is uploading them.