Cluster as a Top Level Resource for Azure Stream Analytics REST API Design

Author: Anthony Pham (anpham) 1/24/2020

Table of Contents

Goals

Outline the details of the REST API that ASA (Azure Stream Analytics) will expose to customers wanting to manage and use “Stream Analytics clusters” as well as changes to any existing REST APIs needed to support the concept of “Stream Analytics clusters” in ASA

Design the REST API in a way that fulfills the VNET/network isolation requirement for JEDI and allow us to extend to fulfill the hardware isolation requirement in the future.

Some scenarios that should be possible as a part of this design (outlined in detail in PM Spec):

- Scale the cluster to expand or shrink capacity
- Share private endpoints within a cluster among multiple streaming jobs
- Associate (link) a streaming job to a cluster
- Configure/update a job to run on a cluster in any subscription they have access to or to multi-tenant cluster
- View list of jobs associated/running in a cluster

Out of scope

How we will integrate the concept of “Stream Analytics clusters” into Synapse

Specific backend implementation details (ex. how are we implementing deployment of the clusters, how are we making calls to Azure Network to create private endpoints, etc.)

Details on how we will meet the hardware isolation requirements.

Clusters

A cluster is a single-tenant, dedicated cluster with VNET support that customers can provision and run their streaming jobs on. This is in contrast to the experience today where customers cannot choose the compute backend their streaming jobs run on and the only choice is to run their streaming jobs on the pre-provisioned multi-tenant cluster.

Clusters will be exposed as a top-level tracked resource in ARM (Azure Resource Manager) at the same level as our other top-level tracked resource, streaming jobs.

Create/Replace Cluster (PUT)

Since provisioning a cluster will take a long time, we will be following the 201 + provisiongState async PUT pattern outlined by the ARM RPC (https://github.com/Azure/azure-resource-manager-rpc/blob/master/v1.0/Addendum.md#creating-or-updating-resources)

批注［**VM1]:** What other states besides 201 created exist?

How many clusters will we limit per subscription?

批注［**KL2]:** Agree, we should have a soft limit per subscription.

Request

Method	Request URI
PUT	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id the cluster should be created in
resourceGroupName	Name of the resource group the cluster should be created under
clusterName	Name of the cluster to create
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Question: Any restrictions on cluster name? The minimum requirement from ARM is: The name cannot include: '<', '>', '%', '&', ':', '\', '?', '/' OR any control characters. The max length is 260 characters. All other characters are allowed.

Request Headers

Common ARM request headers +

Header	On Create	On Replace	Type	Valid Values	Description
If-None-Match	Optional	Not applicable	string	*	Use If-None-Match: * to give PUT create-on semantics – to make create fail if the named resource exists. Without it, PUT replaces the named resource if it exists.
If-Match	Not applicable	Optional	string	GUID string denoting the ETag	Use this header to fail a replace PUT cluster request if the ETag of the cluster does not match the one specified in the header meaning the cluster is in a different state th when you last retrieved it.

Request Body

```
{
   "location": "North US",
   "tags": { "key": "value" },
   "sku" : {
     "name" : "~~ExtraSmall|Small|Medium|Large|~~Default",
     "capacity" : 48
    }
}
```

Property	type	On Create	On Replace	Description

批注［**RB3]:** Either same restrictions as job name or cross check with Synapse.

批注［**KW4R3]:** Portal has been using this Regex for job name “^([\\w-]){3,63}$“ for many years. It’s more restrictive but could avoid potential problems. Let’s continue using it and make RP in sync.

批注［**KL5]:** In this case, what's the expected behavior here without it? We will create a new cluster to replace the old cluster?

批注［**AP6R5]:** If we are not supporting changing VM skus/sizes, then as of right now this would only scale cluster in/out or update tags.

If we support multiple VM skus/sizes, then yes that would require us to delete old cluster and replace it with the new one.

批注［**SA7]:** Would we allow changing location?
This would require deleting and creating cluster, and has the same restrictions as delete (no jobs running etc.)

Somehow I feel that PATCH would work better in this case, and PUT would be create only.

批注［**AP8R7]:** We do not allow changing location. Location is immutable once set. A customer has to delete the current cluster themselves and create it in a different location.

This is similar to other Azure resources and even our streaming job. Location in general is not a mutable property of a resource.

批注［**RB9]:** Current thinking is that we are only going to expose one SKU and as such there is going to option on UI for customers to choose – only capacity. As such, should we also include Default as enum value, and only allow that for now.

批注［**AP10R9]:** Do we want to name the sku “Default”, though? Or just choose a normal value like “Standard” or “Small” to begin with and then we can add more later?

批注［**KI11]:** How does this compare to other services? Synapse for example? EH seems to have pricing tiers?

批注［**AP12R11]:** Do you mean the JSON format/structure? If so, this is the standard JSON format that all RPs need to follow to represent SKU. There are other optional properties that I have not included here since they may not be relevant. Those are tier, size, and family.

The main difference between this and another service I looked at (VMSS) was that in VMSS, they have SKU name as something like Standard_D1_V2 and tier is

批注［**AP13R11]:** For Synapse – I couldn’t find any REST API documentation for their concept of SQL or Spark pools. If you have any pointers, please let me know

location	string	Required	Required. Should be the same value as when specified on create	Region in which to create the cluster Immutable after it is set
tags	dictionary	Optional	Optional	Tags associated with the cluster
sku	object	Required	Required	The SKU of the cluster. This determines the size/capacity of the cluster.
sku.name	string (enum)	Required	Required. Should be the same value as when specified on create? (i.e. we don't support changing VM size)	Specifies the SKU name of the cluster. Acceptable values are ~~ExtraSmall, Small, Medium, and Large~~ Default Could also consider Basic, Standard, Premium, etc.? Immutable after it is set?
sku.capacity	integer	Required	Required	Denotes the number of streaming units the cluster can support Needs to be a discrete set of values (ex. Multiples of 6)? Firaz to give exact list of valid values.

批注［**JB(14]:** From latest discussions I though we wanted to expose 1 SKU for the moment. Should we make the SKU optional for now, and move capacity to top level item?

批注［**AP15R14]:** Not sure what you mean by move capacity to top level item?

The SKU property's JSON format by ARM so we cannot change the structure here. If we want to introduce the concept of SKUs we need to model it this way.

Exposing 1 SKU is fine for now – this design allows for adding more SKUs later on. We should decide on what that 1 SKU's name should be.

批注［**JB(16]:** We won't have all multiple of 6.

Question: Do we need to support a secure/non-secure network mode for cluster? What would we name the property?

Do we need to support MSIs for clusters?

批注［**JB(17]:** I guess we should be able to start without VNET and add it later (was it the question)?

批注［**AP18R17]:** Secure/non-secure network mode meant does the VNET have access to internet ("non-secure") or is blocked off from the internet ("secure mode").

批注［**KL19R17]:** FYI, this is a feature supported by Synapse workspace

Response

Status Code

- 201 (Created) or 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.
- 409 (Conflict) if the cluster is still being provisioned or in a scaling in/out state
- 412 (Precondition Failed) if failed condition specified by If-None-Match or If-Match header.

批注［**KW20]:** In what case will either status code be returned?

Response Headers

Common response headers only

Response Body

Same as GET cluster. Should contain at least the PUT request + additional read-only properties

Update Cluster (PATCH)

Request

Method	Request URI
PATCH	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster that should be updated
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster to update
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

批注［**M(21]:** are we allow change the SKU or other property with same pool name?

Request Headers

Common ARM request headers +

Header	Type	Description
If-Match	string	GUID representing the ETag Use this header to fail a PATCH request if the ETag of the cluster does not match the one specified in the header meaning the cluster is in a different state than when you last retrieved it.

Request Body

Any updatable (non-immutable) property described in the Create/Replace Cluster (PUT) section

Immutable properties are currently **location** and **sku.name**

Users can scale in or out by updating the **sku.capacity** property. In this case, since scaling can be a long operation, we will again use the async PATCH pattern outlined in the ARM RPC.

批注［**KL22]:** do we plan to support it by 6/30?

Restriction: If the current capacity of the cluster is 96 (SUs) and the current running jobs are occupying 48 SUs, then scaling will fail if the user tries to scale down to something lower than 48 SUs. The number they scale to must be able to accommodate the currently running jobs.

Example:

```
{
  "sku" : {
    "capacity" : 96
   }
}
```

Response

Status Code

- 200 (OK) if request completed successfully.
- 202 (Accepted) if request was accepted to complete asynchronously.
- 400 (Bad Request) if the request is not well-formed per the above.
- 404 (Not Found) if the cluster is not found
- 409 (Conflict) if the cluster is not in the right state to be updated (ex. It is still provisioning or scaling)
- 412 (Precondition Failed) if failed condition specified by If-Match header.

批注［**M(23]:** not familar with ARM, just wonder no error regarding permission ? 401/403?

批注［**VM24]:** Is handling this different from handling a 200?

Response Headers

Common response headers only

批注［**KW25]:** ARM: The response headers **MUST** include a Location header that points to a URL where the ongoing operation can be monitored

批注［**AP26R25]:** Yes, I think that falls under common response headers ARM supports

Response Body

Same structure as GET cluster with property values updated as a result of being patched

Get Cluster (GET)

Request

Method	Request URI
GET	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster to retrieve
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster to retrieve
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

批注［**KW27]:** What's the validator for the name? E.g. min/max length

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.
- 404 (Not Found) if the cluster is not found

Response Headers

Common response headers only

Response Body

```
{
  "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/An%20Example%20Streaming%20Pool",
  "name": "An Example Cluster",
  "type": "Microsoft.StreamAnalytics/clusters",
  "location": "Central US",
  "tags": { "key": "value" },
  "sku" : {
    "name" : "ExtraSmall|Small|Medium|Large",
    "capacity" : 96
   }
  "properties": {
    "createdDate": "2020-01-25T01:00Z",
    "clusterId": "B01C67EF-4739-4DDD-9FB2-427EB43DE839",
    "provisioningState": "Succeeded|Failed|Canceled|Creating|Updating|Scaling|etc",
    "capacityUsed": 48,
    "capacityAssigned": 96,
    ~~"streamingJobs": [~~
~~      "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}",~~
~~      "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}"~~
~~    ]~~
  },
  "etag": "F86B9B70-D5B1-451D-AFC8-0B42D4729B8C",
}
```

See Create/Replace Cluster (PUT) Request Body section for an explanation of the writable properties.

Read-only properties seen in the response payload are described below:

Property	type	Description
id	string	The resource id of the cluster. This is a uniquely identifies the cluster within ARM.
name	string	The name of the cluster. Does not uniquely identify the cluster
type	string	The full resource type (typically Resource Provider Namespace + resource type)
properties.createdDate	datetime	The date when this cluster was created Do we also want last updated time?
properties.clusterId	string	Unique identifier for the cluster. Useful for the customer to give us to quickly look up logs on our side.
properties.provisioningState	string (enum)	As per the ARM RPC: The provisioningState field has three terminal states: **Succeeded** , **Failed** and **Canceled**. If the resource returns no provisioningState, it is assumed to be **Succeeded**. Each individual RP is able to define their own transitioning / ephemeral states that are set before the resource reaches these terminal states (e.g. "PreparingVMDisk",

批注 [**KL28]:** do we also want to show all private links as well?

批注 [**AP29R28]:** We should not show that information at this level.

There is an explicit API at the private endpoint level to get that information.

批注 [**KW30]:** Is this property always returned, or we need to specify "$expand" parameter in the request?

批注 [**AP31R30]:** If we go with this route, I would prefer to return this property every time.

批注 [**KW32R30]:** I would also love to see it unless there's major perf impact

批注 [**RB33R30]:** In the current configuration we perhaps are not looking into too many jobs, but going forward we might have clusters that have 1000s of jobs, do we want to return this huge list?

批注 [**M(34R30]:** For Today's GetJobs API, any limit of jobs return? If not, should we keep same behavior?

批注 [**KL35]:** do we also want to show corresponding job status?

批注 [**AP36R35]:** I don't think I see a huge value of having that information here. Also, if users want to manage their jobs, they should be using the job level APIs.

批注 [**AP37R35]:** I just saw your other comment about this property also returning jobs not in a running state in which case this makes more sense to consider.

批注 [**KW38R35]:** +1 for having status in the response

批注 [**M(39R35]:** I agree with Anthony that user should call job level API to get job status. Job status can be changed at any moment, versus job name is static data.

批注 [**KW40R35]:** One scenario Anthony mentioned was that the owner of the streaming pool could lose access to the jobs (e.g. in another sub), so providing status in the response will be a must-have otherwise the owner would have no way to know if the jobs are still running.

批注 [**KW41R35]:** Also, it would be ideal if we can return SU usage info per job.

批注 [**KL42R35]:** How about the scenario where the owner of job lose the permission to access the streaming pool but the job is still running on the pool, do we plan to allow the job owner to change the job status in this case?

批注 [**KL43]:** what would be the value here if the user call the api before any cluster is actually got created but after

批注 [**AP44R43]:** This is the date of when the user created the streaming pool resource not of when we

批注 [**RB45]:** Is this something that's easy to track and the logic of such tracking is clear? If so we might consider,

		"MountingDrives", "SelectingHosts" etc.). Should we do this? If so, what should those states be? This property is utilized in the async PUT protocol to determine if the PUT (provisioning) request successfully completed. Do we need another property for "cluster state"?
properties.streamingUnitsUsed	integer	Represents the number of streaming units currently being used on the cluster. Users can use this to determine how much capacity is being used or is left on the cluster Alternative name: streamingUnitsUsage? Should we duplicate the capacity property as "streamingUnitsCapacity"? Or maybe a "streamingUnitsLeft" property so customers can easily tell how many streaming units are left on their cluster?
properties.capacityAssigned	integer	Represents the sum of the SUs of all streaming jobs associated with the cluster. If all of the jobs were running, this would be the capacity used.
~~**properties.streamingJobs**~~	~~array~~	~~An array of streaming job resource ids. These are the streaming jobs that are currently running on the cluster~~
etag	string	Unique opaque string (generally a GUID) that represents the metadata state of the resource (cluster) **NOTE:** this used to be inside "properties" but most

Question: Is it better to extract some of quota related meta data and make a child proxy-only resource and make an explicit API to get that information? Event Hub dedicated, has something to that effect.

Additionally, would it be better to have a read-only proxy sub-resource called streaming jobs have a specific "List streaming jobs in a cluster" API? This would help with having natural pagination support for listing streaming jobs. Should ask for ARM guidance on this.

We could also add stuff about private endpoint limits? (ex. How many endpoints they have created and what the max number of endpoints they can have)

List Clusters in Subscription (GET)

Request

Method	Request URI
GET	`https://<endpoint>/subscriptions/{subscriptionId}/providers/Microsoft.StreamAnalytics/clusters?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the clusters to retrieve

批注 [**RB46]:** Don't we need at least one that says in progress?

批注 [**AP47R46]:** Yes, we need at least 1 non-terminal state

批注 [**KL48]:** why not just define a clusterState property and put all states there, including clusterProvisionSucceeded, ClusterProvisionFailed, clusterProvisionCancelled and etc.?

批注 [**AP49R48]:** The property needs to be called provisioningState so that it can work with Azure REST API guidelines and the ARM RPC.

批注 [**JB(50]:** streamingUnitsCapacity will be good

批注 [**RB51R50]:** Why not just usedCapacity, should work with capacity property.

批注 [**J(52R50]:** event better!

批注 [**KL53]:** how about jobs failed or jobs not started but associated with the pool?

批注 [**AP54R53]:** We can consider this. My thinking was that it is of more value for the customer to know which jobs are running on the streaming pool.

If a job is simply just associated with the streaming pool I don't see the value of knowing that here.

批注 [**AD55R53]:** I think we should show all the jobs associated with the streaming pool irrespective of job status. What do you think? Wouldn't it help in manageability

批注 [**JB(56]:** do you mean to remove quotas at the job level and set them at the pool level?
We should check what is the reasons for some of our limit (e.g. number of input or output) to see if it's possible.
However since we may want to attach and detach a job from a "pool" maybe we should keep limits as now.

批注 [**AP57R56]:** That's not exactly what I meant. Current design above has the quota information (i.e. streamingUnitsUsed, streamingUnitsCapacity) as information returned in a GET streaming pool API.

The question was more would it make sense to take those properties and expose a specific API under streaming pools to get quota information. So instead of doing a GET streaming pool, you would do a "GET streaming pool quot

批注 [**RB58]:** My vote goes for that. List jobs should be separate API call, similar to private links. Although I understand that private links are tracked and not proxied.

批注 [**KW59R58]:** +1. Paging support would be likely required

批注 [**KL60]:** I think we should expose them. May be at the api for private endpoints below.

api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.

Response Headers

Common response headers only

Response Body

```
{
  “value”: [
  {
    "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/An%20Example%20Streaming%20Pool",
    "name": "An Example Cluster",
    "type": "Microsoft.StreamAnalytics/clusters",
    "location": "Central US",
    "tags": { "key": "value" },
    "sku" : {
      "name" : "ExtraSmall|Small|Medium|Large",
      "capacity" : 96
    }
    "properties": {
      "createdDate": "2020-01-25T01:00Z",
      "clusterId": "B01C67EF-4739-4DDD-9FB2-427EB43DE839",
      "provisioningState": "Succeeded|Failed|Canceled|Creating|Updating",
      "capacityUsed": 48,
      "capacityAssigned": 24,
~~      "streamingJobs": [~~
~~      "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}",~~
~~      "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}"~~
~~      ]~~
    },
    "etag": "71AADF02-1525-404B-8A9C-40AFC8CEFCBF"
  },
  {
    "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/A%20Different%20Streaming%20Pool",
    "name": "A Different Cluster",
    "type": "Microsoft.StreamAnalytics/clusters",
    "location": "Central US",
    "tags": { "key": "value" },
    "sku" : {
      "name" : "ExtraSmall|Small|Medium|Large",
      "capacity" : 96
    }
    "properties": {
      "createdDate": "2020-01-25T01:00Z",
      "clusterId": "967768E2-43D1-494C-80E2-089CDA238D8E",
      "provisioningState": "Succeeded|Failed|Canceled|Creating|Updating",
      "capacityUsed": 48,
      .
      .
    }
  }
  .
  .
  ],
  “nextLink”: “{refererHeaderUrl}?$skipToken={opaqueString}”
}
```

Property	type	Description
value	array	An array of clusters. The structure for each value within the array is the same as Get Cluster response body
nextLink	string	The URL to fetch the next set of clusters (i.e. next page). This property is needed for pagination support. referrerHeaderUrl should be the value from the referer header sent by ARM A value of null means there is no “next page” and the client does not have to continue retrieving clusters.

List Clusters in Resource Group (GET)

Request

Method	Request URI
GET	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the clusters to retrieve
resourceGroupName	Name of the resource group the clusters are under
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.

Response Headers

Common response headers only

Response Body

Same as List Clusters in Subscription response body

List Streaming Jobs in Cluster (POST)

It was recommended by ARM that we use a POST API instead of a GET on a proxy read-only sub-resource for this functionality/scenario. This is because a GET API requires us to confirm to ARM's response body contract and have the resource id of the streaming job be a child of the cluster which is not accurate in this case. POST API action here allows us to define the response body.

It is also strongly recommended to use authorizationActionMapping property in the manifest to map this action to the READ streaming jobs action instead. This will make it so that users using this API must have read streaming job permissions. According to ARM, this is to enforce proper RBAC behavior because this API returns streaming job metadata, not cluster metadata.

批注［KL61]: With this design, does it means that the owner of the cluster may not be able to list all jobs running the cluster due the owner of the cluster may not necessary have permission for all the jobs?

批注［KL62R61]:

批注［AP63R61]: Correct

Request

Method	Request URI
POST	https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/listStreamingjobs?api-version={api-version}

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster the streaming jobs are under
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster the streaming jobs are under
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.

Response Headers

Common response headers only

Response Body

```
{
  “value”: [
  {
    "id":
"/subscriptions/{id}/resourceGroups/{group}/providers/microsoft.streamAnalytics/streaming
jobs/A%20%Filter%20Sample",
    "streamingUnits": 6,
    "jobState": "Running",
  },
  {
    "id":
"/subscriptions/{id}/resourceGroups/{group}/providers/microsoft.streamAnalytics/streaming
jobs/Another%20%Filter%20Sample",
    "streamingUnits": 1,
    "jobState": "Stopped",
      .
      .
    }
  }
  .
  .
  ],
  “nextLink”: “{refererHeaderUrl}?$skipToken={opaqueString}”
}
```

All properties below are read-only

Property	type	Description
value	array	An array of streaming jobs associated with the cluster
value[n].id	string	Resource Id of the streaming job
value[n].streamingUnits	Integer	The number of streaming units that are used by the streaming job
Value[n].jobState	string (enum)	Indicates the current execution state of the streaming job
nextLink	string	The URL to fetch the next set of streaming jobs (i.e. next page). This property is needed for pagination support. referrerHeaderUrl should be the value from the referer header sent by ARM A value of null means there is no “next page” and the client does not have to continue retrieving streaming jobs.

Delete Cluster (DELETE)

~~A delete cluster request will implicitly stop all running or starting streaming jobs in that cluster. Once the request completes, the cluster will no longer and exist and all of the streaming jobs that were running on it will be in stopping or stopped state.~~

Question: Should delete cluster only complete after it has confirmed all of the running jobs are stopped?

批注［**JB(64]:** This will be safer.

批注［**KI65R64]:** +1

A delete cluster request should error out if jobs are still running on the cluster. Jobs should be stopped and unlinked before making a delete cluster request.

批注［**KI66]:** +1

批注［**KL67]:** Can we also add all private links has been deleted? This would make us life easier for managing the life cycle of private links.

批注［**RB68]:** Anthony, can we also discuss soft delete. I have a feeling that soft delete is something that's going to come as a requirement pretty soon.

批注［**AP69R68]:** Good call. We should take that into account

批注［**AP70R68]:** ARM said they do not have any concrete guidance on this yet

Need to confirm with ARM that deleting a cluster would also delete private endpoints. – Confirmed that yes we should also delete private endpoints. We should look at resourceDeletionPolicy = Cascade in RP Manifest

Request

Method	Request URI
DELETE	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster to delete
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster to delete
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

批注［**KL71]:** shall we add status code for conflict operation? e.g. another delete request comes while the operation is on-going

批注［**AP72R71]:** A delete should generally always trump whatever other operation is on-going

- 200 (OK) or 204 (No Content) if request completed successfully.
- 202 (Accepted) if request was accepted to complete asynchronously.
- 400 (Bad Request) if the request is not well-formed per the above.
- 412 (Precondition Failed) if failed condition specified by If-Match header.

Response Headers

Common response headers only

批注［**KW73]:** Per ARM spec, please make sure to include a **Location** header to monitor the operation.

批注［**AP74R73]:** Right, location header falls under the common response headers ARM supports

Response Body

Empty

Remove Streaming Job (POST) – Not on initial release

It was recommended by ARM to not have this API on initial release and wait for customer feedback to see if this API is really needed.

This API will allow users to stop streaming jobs that are currently running on the cluster and disassociate the streaming jobs from the cluster (i.e. null out the cluster property on the job). Clusters do not restrict the subscription id or resource group of the streaming job that is being scheduled. As a result, streaming jobs that initially had access to be scheduled on a cluster may have that access revoked. Managers of the cluster can then call this API to stop those streaming jobs that do not have access anymore without fear of having those streaming jobs being scheduled on the cluster again. Without this, the only other way would be to delete the cluster and create a new one which is not an ideal user experience.

ARM may not agree with our solution here so we may need to consider options described in aka.ms/azureiamonboarding which is part of their official RBAC guidance.

Request

Method	Request URI
`POST`	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/removeStreamingJob?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

```
{
  "streamingJobResourceId":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft
.StreamAnalytics/streamingjobs/MyStreamingJob"
}
```

Property	type	Required	Description
streamingJobResourceId	string	Yes	The resource id of the streaming job you want to remove

Response

Status Code

- 200 (OK) if request completed successfully.
- 202 (Accepted) if request was accepted to complete asynchronously.
- 400 (Bad Request) if the request is not well-formed per the above (ex. If the streaming job specified is not running on the cluster).
- 404 (Not Found) if the cluster is not found

Response Headers

Common response headers only

Response Body

Empty

~~Stop Streaming Job (POST) –~~ Not needed. Replaced with "Remove Streaming Job"

This API will allow users to stop streaming jobs that are currently running on the cluster. Clusters do not restrict the subscription id or resource group of the streaming job that is being scheduled. As a result, streaming jobs that initially had access to be scheduled on a cluster may have that access revoked. Managers of the cluster can then call this API to stop those streaming jobs that do not have access anymore without fear of having those streaming jobs being scheduled on the cluster again. Without this, the only other way would be to delete the cluster and create a new one which is not an ideal user experience.

Request

Method	Request URI
POST	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/stopStreamingJob?api-version={api-version}`

批注［**JB(75]:** Do you mean stop?

批注［**AP76R75]:** Yup! Fixed

Parameters

Parameter	Description
subscriptionId	Subscription id of the cluster
resourceGroupName	Name of the resource group the cluster is under
clusterName	Name of the cluster
api-version	A supported API version for ASA Clusters (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

```
{
  "streamingJobResourceId":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft
.StreamAnalytics/streamingjobs/MyStreamingJob"
}
```

Property	type	Required	Description
streamingJobResourceId	string	Yes	The resource id of the streaming job you want to stop

批注［KW77]: Shall we support stopping multiple jobs? User might want to perform bulk operation when looking at the list of jobs.

批注［KL78R77]: +1

批注［AP79R77]: We should definitely consider that. In that case, I can change the request body to take an array of resource ids

批注［KW80R77]: You mean request body?

批注［RB81R77]: I want to know what the scenario is from user perspective. Meaning, if we have single job API they can automate many scenarios. In other words if we are considering an array, why not also consider delate all jobs API.

Response

Status Code

- 200 (OK) if request completed successfully.
- 202 (Accepted) if request was accepted to complete asynchronously.
- 400 (Bad Request) if the request is not well-formed per the above (ex. If the streaming job specified is not running on the cluster).
- 404 (Not Found) if the cluster is not found

批注［KL82]: also for job not found

批注［AP83R82]: For that I put that as a 400 Bad Request.

404 Not Found usually refers to the resource in the url path not being found. It doesn't usually point to stuff in the request body.

Response Headers

Common response headers only

Response Body

Empty

Start/Stop, Reallocate/Deallocate, PowerOn/PowerOff, TurnOn/TurnOff Cluster? (POST)

Do we need to actions to be able to "deallocate" a cluster similar to how you can turn on/off a virtual machine? In a deallocated state, the cluster would not be usable (i.e. you cannot start jobs on it), but the customer would not be billed for the cluster **and** the private endpoints would still be valid (they would still be billed for the private endpoint). Deallocate in this case may mean we just shut down the VMs in the SF clusters (or we may choose to just delete these resources) so that we don't incur costs ourselves. Later, if the user decides to use the cluster again, they can reallocate or turn it back on in which case they would begin to be billed again. Reallocating in this case could be faster than provisioning a new cluster (if all we had to do was turn the VMs on again); otherwise, it would just mean provisioning a new SF cluster and reuse the previous VNET.

批注［KI84]: Seems useful, but probably not P0?

批注［RB85R84]: +1

批注［JB(86]: Interesting concept, it's similar to the "pause" function on DW, and may be interesting for dev/test scenarios.

This could be useful if we foresee scenarios where a customer might not be using the cluster for a month so they don't want to pay for it, but they don't want to go through the hassle of deleting the cluster and then provisioning it a month later and also have to set up their private endpoints again.

Potential Database Schema

We will need to create a new SQL Table to hold information about the Stream Analytics cluster. Potential table schema below. Schema is subject to change based on implementation details that might need to store more internal tracking information that is not exposed to customers

```
CREATE TABLE [dbo].[Cluster]
(
    [Id] UNIQUEIDENTIFIER NOT NULL PRIMARY KEY,
    [ResourceGroupName] ResourceGroupNameType2 NOT NULL,
    [Name] ResourceNameType NOT NULL,
    [SubscriptionId] INT NOT NULL,
    [SFClusterId] UNIQUEIDENTIFIER NOT NULL,
    [LastUserAction] StatusType NOT NULL,
    [Status] StatusType NOT NULL,
    [Location] LocationType NOT NULL,
    [Tags] NVARCHAR(MAX) NOT NULL,
    [Sku] SkuType NULL,
Sku + capacity
    [Data] NVARCHAR(MAX) NOT NULL, -- Holds bulk of request/response body which is
nothing at the moment but could be used later if we extend it
    [OperationId] OperationIdType NULL,
    [AzureAsyncOperationId] OperationIdType NULL,
    [Etag] UNIQUEIDENTIFIER NOT NULL,
    [CreatedDateTime] DATETIME NOT NULL,
    [LastUpdatedDateTime] DATETIME NOT NULL,
    [IsDeleted] bit NOT NULL,
    FOREIGN KEY ([SubscriptionId]) REFERENCES [Subscription](Id)
    FOREIGN KEY ([SFClusterId]) REFERENCES [SFClusterInfo](Id)
)
```

批注［**M(87]:** Add Capacity please.

Private Endpoints

A cluster is created with its own VNET and naturally does not have access to customer resources that are behind their own VNET. Therefore, we need a mechanism in which customers can grant the cluster access to their resources. Azure already has a concept of Azure Private Endpoints to allow a VNET secure access to another service/resource behind a different VNET. We will be using the same concept and modeling our REST API after the official Azure Private Endpoints resource for consistency.

Private endpoints will be exposed as a nested, tracked resource in ARM (Azure Resource Manager) under a cluster. It needs to be a tracked resource because it will be billed and ARM requires billable resources to support tags which can only be done if it is a tracked resource.

Create/Replace Private Endpoint (PUT)

Will creation of Private Endpoint be synchronous or asynchronous? Need more information from Ke to make a call

批注［**KL88]:** Let's make it synchronous here. I will share a design doc about the workflow.

How many private endpoints will we limit per cluster? Should there be a global subscription limit as well?

批注［**KL89]:** this limit will be the same as the number of private endpoints for a vnet. It's 1000 currently. https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#private-link-limits

批注［**RB90R89]:** Did we not have another limit which was per subscription? If so, our limit can be less than 1000.

批注［**KL91R89]:** are you referring to private dns zone here? each private dns zone could hold 25000 records and each record will be used by one private endpoint. We should be able to support 1000 private private link here.

批注［**KL92]:** technically, we may not have to put a restriction here. However, we might want to warn the customers if they have too many private links created which might result in high bill

I have chosen to version Private Endpoints with a different API version set than Clusters. This is because we will have to declare them separately in the RP Manifest so I thought it made sense for them to have different API version sets. This also allows us to make updates to Cluster APIs without affecting the api version of Private Endpoints and vice-versa. Thoughts?

Need to clarify with ARM – can users create subresource while parent resource is still in a non-terminal provisioning state?

Request

Method	Request URI
PUT	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/privateEndpoints/{privateEndpointName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id the private endpoint should be created in
resourceGroupName	Name of the resource group the private endpoint should be created under
clusterName	Name of the cluster the private endpoint should be created under
privateEndpointName	Name of the private endpoint to create
api-version	A supported API version for ASA Private Endpoints (ex. 2020-01-01)

批注［**KW93]:** What's the validator for the name? E.g. min length

Question: Any restrictions on private endpoint name? The minimum requirement from ARM is: The name cannot include: '<', '>', '%', '&', ':', '\', '?', '/' OR any control characters. The max length is 260 characters. All other characters are allowed.

批注［**KL94]:** I got below when try to create a private endpoint from Azure portal. Our rule should be no less restrictive than below.
The name must be between 1 and 80 characters.
The name must begin with a letter or number, end with a letter, number or underscore, and may contain only letters, numbers, underscores, periods, or hyphens.

Request Headers

Common ARM request headers +

Header	On Create	On Replace	Type	Valid Values	Description
If-None-Match	Optional	Not applicable	string	*	Use If-None-Match: * to give PUT create-only semantics – to make create fail if the named resource exists. Without it, PUT replaces the named resource if it exists.
If-Match	Not applicable	Optional	string	GUID string denoting the ETag	Use this header to fail a replace PUT private endpoint request if the ETag of the private endpoint does not match the one specified in the header meaning the private endpoint is in a different state than when you last retrieved it.

Request Body

批注［**KL95]:** We may need add properties.requestMessage in https://docs.microsoft.com/en-us/rest/api/virtualnetwork/privateendpoints/createorupdate#privatelinkserviceconnection
To our request body if NRP end confirms that's the right way for user to know this is valid request. We're still following with network team about this.

Note that there is **no sku** property. Since the SKU concept does not make sense currently for private endpoints (Azure Network also does not expose a concept of SKU for private endpoints), we will not support a sku property at this time. Customers are currently billed a static, well-defined amount per private endpoint.

```
{
  "location": "eastus2euap",
  "tags": { "key": "value" },
  "properties": {
    "privateLinkServiceConnections": [
      {
        "properties": {
          "privateLinkServiceId":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateLinkService
s/testPls",
          "groupIds": [
            "groupIdFromResource"
          ]
        }
      }
    ]
  }
}
```

Property	type	On Create	On Replace	Description
location	string	Required. Should be the same value as the location of the cluster	Required. Should be the same value as when specified on create	Region in which to create the private endpoint Immutable after it is set
tags	dictionary	Optional	Optional	Tags associated with the private endpoint
properties	object	Required	Required	Bag of properties specifically related to private endpoints
properties.privateLinkServiceConnections	array	Required	Required	A grouping of information about the connection to the remote resource.
properties.privateLinkServiceConnections.properties	object	Required	Required	Bag of properties defining a privatelinkServiceConnection
properties.privateLinkServiceConnections.properties.privateLinkServiceId	string	Required	Required	The resource id of the resource we want to privately connect to **Azure Network REST API documentation description:** “The resource id of private link service.”
properties.privateLinkServiceConnections.properties.groupIds	array	Required	Required	The subresource(s) of the privateLinkServiceId to connect to. For example, if you specify a storage account resource id in privateLinkServiceId, potential groupIds (subreseources) to connect to are blob, tables, queues, files, etc.

				We should only support the subresources that ASA supports. **Azure Network REST API documentation description:** The ID(s) of the group(s) obtained from the remote resource that this private endpoint should connect to. TODO: Need to figure out what the actual supported values are as Azure Network's documentation does not publicly state this.

批注［**RB96]:** +1 Ke, can you please look into this?

批注［**KL97R96]:** It's listed in the subresource column from https://docs.microsoft.com/en-us/azure/private-link/private-endpoint-overview#dns-configuratio.

Question: What does it mean to do a replace on a private endpoint? What happens if I set up a private endpoint to a storage account and then do a PUT on that same private endpoint but specify SQL Server instead?

批注［**KL98]:** I am proposing all data in the request body is immutable.

批注［**RB99R98]:** Valid question, we probably don't need Patch on this.

批注［**KL100R98]:** Patch might be useful for tags here.

Comparison with Azure Private Endpoint's Request Body

Below is the request body a user would make if they were making a request to Azure Private Endpoints normally taken from https://docs.microsoft.com/en-us/rest/api/virtualnetwork/privateendpoints/createorupdate#create-private-endpoint

The proposed request body in the previous section is simply a subset of the original request body below.

The "subnet" section is not applicable in our scenario because the subnet is the subnet of the cluster's VNET which only ASA would know so we would fill that out on behalf of the user.

~~We also decided to not have the "privateLinkServiceConnections" array. In normal Azure Network Private Endpoints, 1 private endpoint maps a subnet to multiple resources. This is probably due to the fact that customers can create multiple private endpoints for multiple subnets. However, in our scenario, the private endpoint you are creating is for the same subnet (the cluster's VNET subnet). In that case, we thought it would be easier to manage private endpoints with a 1:1 mapping (1 private endpoint per resource that needs to be accessed). Creating 1 private endpoint under a cluster that would handle connections to all of your resources did not seem natural/intuitive.~~

~~**However**, since this forces customers to create multiple private endpoints, this increases the cost for the customer as opposed to in Azure Network, where it seems possible to create 1 private endpoint to go to multiple resources in which case they would only pay for 1 private endpoint instead of multiple. If so, this acceptable?~~

批注［**RB101]:** We should follow up on this section.

批注［**KL102R101]:** I will follow up on this.

批注［**KL103R101]:** Mario confirmed that this is no plan for networking side to support more than 1 PLS be associated with 1 PE

```
{
   "location": "eastus2euap",
   "properties": {
     "privateLinkServiceConnections": [
       {
         "properties": {
           "privateLinkServiceId":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateLinkService
s/testPls",
           "groupIds": [
             "groupIdFromResource"
           ],
           "requestMessage": "Please approve my connection."
         }
       }
     ],
     "subnet": {
       "id":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/virtualNetworks/my
Vnet/subnets/mySubnet"
     }
   }
}
```

Response

Status Code

- 201 (Created) or 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.
- 412 (Precondition Failed) if failed condition specified by If-None-Match or If-Match header.

Response Headers

Common response headers only

Response Body

Same as GET Private Endpoint. Should contain at least the PUT request + additional read-only properties

Update Private Endpoint (PATCH)?

Azure Network does not support PATCH for a private endpoint. Does that mean we can do the same? Need to clarify with ARM team as my understanding is PATCH is a required API to support for any tracked ARM resource.

批注［**RB104]:** Yes, let’s not have a patch.

批注［**KL105R104]:** we may have to support patch for tag field.

Get Private Endpoint (GET)

Request

Method	Request URI
GET	https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/privateEndpoints/{privateEndpointName}?api-version={api-version}

Parameters

Parameter	Description
subscriptionId	Subscription id of the private endpoint to retrieve
resourceGroupName	Name of the resource group the private endpoint is under
clusterName	Name of the cluster the private endpoint is under
privateEndpointName	Name of the private endpoint to retrieve
api-version	A supported API version for ASA Private Endpoints (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.
- 404 (Not Found) if the private endpoint is not found

Response Headers

Common response headers only

Response Body

```
{
  "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/{clusterName}/privateEndpoints/An%20Example%20Private%20Endpoint"
,
  "name": "An Example Private Endpoint",
  "type": "Microsoft.StreamAnalytics/clusters/privateEndpoints",
  "location": "eastus",
  "properties": {
    "createdDate": "2020-01-25T01:00Z",
    "provisioningState": "Succeeded",
    "privateLinkServiceConnections": [
      {
        "properties": {
          "privateLinkServiceId":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateLinkServices/
testPls",
          "groupIds": [
            "groupIdFromResource"
          ],
          "requestMessage": "Please approve my connection.",
          "privateLinkServiceConnectionState": {
            "status": "Approved",
            "description": "Auto-approved",
            "actionsRequired": "None"
          }
        }
      }
    ]
  },
  "etag": "F86B9B70-D5B1-451D-AFC8-0B42D4729B8C",
}
```

See Create/Replace Private Endpoint (PUT) Request Body section for an explanation of the writable properties.

Read-only properties seen in the response payload are described below (some description text copied from Azure Network Private Endpoints REST API documentation):

Property	type	Description
id	string	The resource id of the private endpoint This is a uniquely identifies the private endpoint within ARM.
name	string	The name of the private endpoint. Does not uniquely identify the private endpoint
type	string	The full resource type (typically Resource Provider Namespace + resource type)
properties.createdDate	string	The date when this private endpoint was created Azure Private Endpoints does not expose a createdDate read-only property. Do we want to stay consistent with them or are we okay exposing this extra information? Do we also want last updated time?
properties.provisioningState	string (enum)	As per the ARM RPC: The provisioningState field has three terminal states: **Succeeded** , **Failed** and **Canceled**. If the resource returns no provisioningState, it is assumed to be **Succeeded**. Each individual RP is able to define their own transitioning / ephemeral states that are set before the resource reaches these terminal states (e.g. "PreparingVMDisk", "MountingDrives", "SelectingHosts" etc.). This property is utilized in the async PUT protocol to determine if the PUT (provisioning) request successfully completed.
properties.privateLinkServiceConnections .properties.privateLinkServiceConnectionState	object	A collection of read-only information about the state of the connection to the private remote resource.
properties.privateLinkServiceConnections .properties.privateLinkServiceConnectionState.status	string	Indicates whether the connection has been Approved/Rejected/Removed by the owner of the remote resource/service.
properties.privateLinkServiceConnections .properti	string	The reason for approval/rejection of the connection.

批注［**KL106]:** Should this be the timestamp when we get the request and save the information into the db instead of using the timestamp from the real private link resource?

es.privateLinkServiceConnectionState.description		
properties.privateLinkServiceConnections .properties.privateLinkServiceConnectionState.actionsRequired	string	A message indicating if changes on the service provider require any updates on the consumer. Do we need this property?
etag	string	Unique opaque string (generally a GUID) that represents the metadata state of the resource (private endpoint)

批注［**KL107]:** Yes, we may need indicate customers to approve the request

Comparison with Azure Private Endpoint’s Response Body

Below is the request body a user would make if they were making a request to Azure Private Endpoints normally taken from https://docs.microsoft.com/en-us/rest/api/virtualnetwork/privateendpoints/get#get-private-endpoint

Similar with the request body, our response body is a subset of the original Azure Private Endpoint’s response body. We remove all of the properties that are not relevant to the customer in this case such as subnet and network interface information since the subnet should be implied to be the cluster’s subnet.

```
{
  "name": "testPe",
  "id":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateEndpoints/t
estPe",
  "type": "Microsoft.Network/privateEndpoints",
  "location": "eastus",
  "properties": {
    "provisioningState": "Succeeded",
    "privateLinkServiceConnections": [
      {
        "properties": {
          "privateLinkServiceId":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateLinkService
s/testPls",
          "groupIds": [
            "groupIdFromResource"
          ],
          "requestMessage": "Please approve my connection.",
          "privateLinkServiceConnectionState": {
            "status": "Approved",
            "description": "Auto-approved",
            "actionsRequired": "None"
          }
        }
      }
    ],
    "manualPrivateLinkServiceConnections": [],
    "subnet": {
      "id":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/virtualNetworks/my
Vnet/subnets/mySubnet"
    },
    "networkInterfaces": [
      {
        "id":
"/subscriptions/subId/resourceGroups/rg1/provders/Microsoft.Network/networkInterfaces/t
estPe.nic.abcd1234"
      }
    ]
  }
}
```

List Private Endpoints in Cluster (GET)

Request

Method	Request URI
GET	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/privateEndpoints?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the private endpoints to retrieve
resourceGroupName	Name of the resource group the private endpoints are under
clusterName	Name of the cluster the private endpoints are under
api-version	A supported API version for ASA Private Endpoints (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) if request completed successfully.
- 400 (Bad Request) if the request is not well-formed per the above.

Response Headers

Common response headers only

Response Body

```
{
  “value”: [
  {
    "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/{clusterName}/privateEndpoints/An%20Example%20Private%20Endpoint"
,
    "name": "An Example Private Endpoint",
    "type": "Microsoft.StreamAnalytics/clusters/privateEndpoints",
    "location": "eastus",
    "properties": {
      "createdDate": "2020-01-25T01:00Z",
      "provisioningState": "Succeeded",
      "privateLinkServiceId":
"/subscriptions/subId/resourceGroups/rg1/providers/Microsoft.Network/privateLinkServices/
testPls",
      "groupIds": [
        "groupIdFromResource"
      ],
      "privateLinkServiceConnectionState": {
        "status": "Approved",
        "description": "Auto-approved",
        "actionsRequired": "None"
      }
    },
    "etag": "F86B9B70-D5B1-451D-AFC8-0B42D4729B8C",
  },
  {
    "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.S
treamAnalytics/clusters/{clusterName}/privateEndpoints/A%20Different%20Private%20Endpoint
",
    "name": "A Different Private Endpoint",
    "type": "Microsoft.StreamAnalytics/clusters/privateEndpoints",
    "location": "eastus",
    "properties": {
      "createdDate": "2020-01-25T01:00Z",
      "provisioningState": "Succeeded",
      .
      .
    }
  }
  .
  .
  ],
  “nextLink”: “{refererHeaderUrl}?$skipToken={opaqueString}”
}
```

Property	type	Description
value	array	An array of private endpoints. The structure for each value within the array is the same as [Get Private Endpoint response body](#)
nextLink	string	The URL to fetch the next set of private endpoints (i.e. next page). This property is needed for pagination support. referrerHeaderUrl should be the value from the referer header sent by ARM A value of null means there is no “next page” and the client does not have to continue retrieving private endpoints.

Delete Private Endpoint (DELETE)

Request

Method	Request URI
DELETE	`https://<endpoint>/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.StreamAnalytics/clusters/{clusterName}/privateEndpoints/{privateEndpointName}?api-version={api-version}`

Parameters

Parameter	Description
subscriptionId	Subscription id of the private endpoint to delete
resourceGroupName	Name of the resource group the private endpoint is under
clusterName	Name of the cluster the private endpoint is under
privateEndpointName	Name of the private endpoint to delete
api-version	A supported API version for ASA Private Endpoints (ex. 2020-01-01)

Request Headers

Common ARM request headers only

Request Body

Empty

Response

Status Code

- 200 (OK) or 204 (No Content) if request completed successfully.
- 202 (Accepted) if request was accepted to complete asynchronously.
  - Not sure if this API will be async or sync yet. Need information from Ke to make a call
- 400 (Bad Request) if the request is not well-formed per the above.
- 412 (Precondition Failed) if failed condition specified by If-Match header.

批注 [**KL108]:** It will be async.

Response Headers

Common response headers only

Response Body

Empty

批注 [**KW109]:** How does the client track the async operation?

Potential Database Schema

We will need to create a new SQL Table to hold information about private endpoints. Potential table schema below. Schema is subject to change based on implementation details that might need to store more internal tracking information that is not exposed to customers

```
CREATE TABLE [dbo].[PrivateEndpoints]
(
    [Id] UNIQUEIDENTIFIER NOT NULL PRIMARY KEY,
    [ResourceGroupName] ResourceGroupNameType2 NOT NULL,
    [ClusterId] UNIQUEIDENTIFIER NOT NULL,
```

批注 [**KL110]:** Can we have a column for PLS ARM id and PE name we send to NRP? We can skip the PE name column if we just use the same name file here.
When we implement the data changes: We would need have a constraint to avoid the same PLS with the same sub resource type been created on the same cluster multiple times, this might cause DNS resolution conflict. Simplar logic should also add to the implementation of rest API.

批注 [**KL111R110]:** Also, we would need store sub resource type which group id in a seperate column

```
    [Name] ResourceNameType NOT NULL,
    [SubscriptionId] INT NOT NULL,
    [Location] LocationType NOT NULL,
    [Tags] NVARCHAR(MAX) NOT NULL,
    [ApprovalStatus] StatusType NOT NULL, -- Could be useful for Ke to see if PE is
approved or not
    [PrivateLinkServiceId] NVARCHAR(MAX) NOT NULL,
    [GroupId] NVARCHAR(MAX) NOT NULL,
    [Data] NVARCHAR(MAX) NOT NULL, -- Holds bulk of request/response body
    [OperationId] OperationIdType NULL,
    [AzureAsyncOperationId] OperationIdType NULL,
    [Etag] UNIQUEIDENTIFIER NOT NULL,
    [CreatedDateTime] DATETIME NOT NULL,
    [LastUpdatedDateTime] DATETIME NOT NULL,
    [IsDeleted] bit NOT NULL,
    FOREIGN KEY ([SubscriptionId]) REFERENCES [Subscription](Id)
    FOREIGN KEY ([ClusterId]) REFERENCES [Cluster](Id)
)
```

批注［KL112]: Yes, this status would be helpful here. Let's follow the naming convention from networking and name it ConnectionStatus. We would also need a column to indicate the status from our side, e.g, successful, deleting and etc.

批注［KL113R112]: I will add schema related with private DNS zone in the design for private links management

批注［KL114]: can we have an example for this? Is it used for holding response from NRP?

Existing APIs that need to be updated

Job Object Model

To enable users to choose which cluster their streaming jobs will run on, we will need to update the streaming job object model to support a new property called "cluster". This will also indicate to the service where to run test connection and sample input. The value of the property will be the resource id of the cluster the user wants the streaming job to run on. Another requirement is that the cluster must be in the **same location** as the job (i.e. if your job is in West US, you cannot specify the cluster resource id of a cluster that is in Southeast Asia). A null value for this property indicates that the streaming job should run on the multi-tenant cluster (i.e. the default behavior today). See sample JSON below on what the property would look like in the request/response body.

```
{
  "id":
"/subscriptions/{id}/resourceGroups/{group}/providers/microsoft.streamAnalytics/streaming
jobs/filterSample",
  "name": "A Filter Sample",
  "type": "Microsoft.StreamAnalytics/jobs",
  "location": "North US",
  "tags": { "key": "value" },

  "properties": {
    "sku": {
      "name": "{ standard }"
    },
    "provisioningState": "Succeeded|Failed|Canceled|Creating|Updating",
    "jobState": "Not Started|Pending|Starting|Running|Stopping|Stopped|Failed",
    "outputStartMode" : "JobStartTime | CustomTime",
    "outputStartTime": "2014-07-03T01:00Z",
    "eventsOutOfOrderPolicy": "adjust|drop",
    "eventsOutOfOrderMaxDelayInSeconds": 10,
    "eventsLateArrivalMaxDelayInSeconds": -1,
    "cluster": "/subscriptions/409468b9-1047-4f2c-b98b-
9245c3d1448d/resourceGroups/IoTResources/providers/Microsoft.StreamAnalytics/clusters/MyC
luster",
    "inputs": [],
```

批注［KW115]: Do we support move subscription/resourceGroup ARM operation on StreamingPool?

批注［AP116R115]: Yes we do. Good point, we may need to update the property as well in that case.

```
    "outputs": [],
    "transformation": {},
    "functions": [],
    "etag": "71AADF02-1525-404B-8A9C-40AFC8CEFCBF"
  }
}
```

Running streaming jobs on a cluster that is in a different subscription

In order to support the scenario where a user can start a streaming job in subscription 1 on a cluster in subscription 2, we need a way to verify that the user has access to the cluster in subscription 2 otherwise anyone who knows the resource id of a cluster can potentially start a streaming on it.

The current proposed solution is to make use of ARM RBAC Linked Access Checks. This feature puts the onus on ARM to do the RBAC check before proxying the request to our RP. In our RP Manifest, we specify which action needs an RBAC check, which property contains the resource id to do the check on, and the action on that resource that the user is required to have to in order to set that property. **Note:** we have not yet verified nor confirmed that this solution will work. At this point this is simply an approach we came up with based on our reading of ARM documentation. POC still needs to be done.

In the above proposal, the RBAC check would be done on write operations on the streaming job (i.e. PUT/PATCH) on the cluster property since those are the APIs where the property is set. We would also need to create a custom action with the scope of “start a starting job on clusters” (unclear at the moment on how to do this – is it simply adding that action in the Operations API even if there is no backing API behind it?) which would be the permission the user is required to have on the cluster during the RBAC check.

Need to clarify with ARM on how to null out or set this property to a non-resource id value.

~~**Concerns about this approach:** Since the RBAC check is only done on PUT/PATCH of the streaming job, this implies that once it is set on the job, the job can forever be scheduled on that cluster unless someone overwrites the property or the job is deleted. If the original user who set the property lost access, we expose a Stop Streaming Job API on the cluster so that managers of the cluster can stop the streaming job after the access has been revoked if necessary. **However**, there is nothing stopping users from subscription 1 to simply start the streaming job again. Since start job does not do the RBAC check, the streaming job will start on the same cluster again.~~

~~Does this mean it may be a better design to have users specify which cluster to run on at the API/operations level (i.e. start job, test connection, sample input) as opposed to the streaming job level? If the RBAC check was performed at start job, test connection and sample input instead, then the streaming job would not be allowed to be scheduled on the cluster in subscription 2 once the user lost access. This would also imply that start job is the API that sets the cluster property on the streaming job and it is otherwise a read-only property. However, we did not originally go with this approach since it might be a bad user experience to have to continuously specify a cluster every time you do a test connection or sample input where you might iterate multiple times.~~

Potentially addressed above concern by having Remove Streaming Job (POST) API.

批注［YZ117]: I prefer the StreamingPool is in the API path instead of the JSON body. It’s a clearer ownership and ARM can easily validate user’s permission on the StreamingPool entity.
It’s easier for authoring for customer too, i.e. no need to construct ARM id for the streaming pool in JSON body (if an advanced user do direct REST call etc.).

批注［YZ118R117]: Although Synapse is not in scope, forward-thinking, it’s easier to construct API call using StreamingPool than construct JSON payload with StreamingPool in it.

批注［AP119R117]: Either option will require user to specify the streaming pool in the JSON body. When I said “at the API/operations level” I meant the JSON body of those APIs.

There is no option to specify streaming pool in the URL because that does not really make sense and we would not able to verify the user has access to that streaming pool.

批注［AP120R117]:

批注［RB121]: What would be the API or mechanism to go back to ARM and ask if the user has access to the resource? If you’ll be meeting with ARM folks, can you please have tack about this too?

Move Resources API

For this API, we receive a list of resource ids in the request body (see format below). Only tracked resources can be specified in the request body. Therefore, proxy-only resources such as inputs, outputs, transformations, etc. and are implicitly moved when a streaming job is moved. However, since private endpoints are a tracked resource and are nested under clusters which is another tracked resource, the logic becomes a little different. In this case, the resource id of the cluster **and** the resource ids of **all** private endpoints under that cluster must be explicitly specified in the request body in order to be moved. It does not make sense to move a cluster without the underlying private endpoints and vice versa. Therefore, ARM requires that a user must explicitly specify all of the resource ids of the tracked resources in that resource hierarchy.

```
{
   "targetResourceGroup": "/subscriptions/{targetId}/resourceGroups/{targetName}",
   "resources":
   [
     "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}",
     "/subscriptions/{id}/resourceGroups/{source}/providers/{namespace}/{type}/{name}"
   ]
}
```

Get Quotas API

Our Quotas API currently returns information about SU quota on the subscription. We need to extend it to provide information about subscription limits for things related to clusters. For example, the max number of clusters you can have on a subscription or the max number of private endpoints you can create (if we set a global subscription limit on it).

批注［**KL122]:** How about we also display the quota for different number of input/output types supported in a cluster here

```
{
  "value": [
    {
      "id":
"/subscriptions/{subscriptionId}/providers/Microsoft.StreamAnalytics/locations/{location}
/quotas/StreamingUnits",
      "name": "StreamingUnits",
      "type": "Microsoft.StreamAnalytics/quotas",
      "properties": {
        "maxCount": 12,
        "currentCount": 6
      }
    },
    {
      "id":
"/subscriptions/{subscriptionId}/providers/Microsoft.StreamAnalytics/locations/{location}
/quotas/StreamingUnits",
      "name": "Clusters",
      "type": "Microsoft.StreamAnalytics/quotas",
      "properties": {
        "maxCount": 50,
        "currentCount": 16
      }
    },
    .
    .
```

批注［**RB123]:** Capacity/usedCapacity maybe?

```
  ]
}
```

Operations API

Currently our RP exposes an operations metadata API that allows users to understand what kind of operations they can perform against our RP/what operations our RP supports. Today, that consists mainly of operations for a streaming job and metadata about the metrics we expose (example snippets below).

We will need to update this API to also return possible operations on clusters and private endpoints.

```
{
  "value": [
    {
      "name": "Microsoft.StreamAnalytics/streamingjobs/Delete",
      "display": {
        "provider": "Microsoft Azure Stream Analytics",
        "resource": "Stream Analytics Job",
        "operation": "Delete Stream Analytics Job",
        "description": "Delete Stream Analytics Job"
      }
    },
    {
      "name": "Microsoft.StreamAnalytics/streamingjobs/providers/Microsoft.Insights/metri
cDefinitions/read",
      "display": {
        "provider": "Microsoft Azure Stream Analytics",
        "resource": "The metric definition of streamingjobs",
        "operation": "Read streamingjobs metric definitions",
        "description": "Gets the available metrics for streamingjobs"
      },
      "properties": {
        "serviceSpecification": {
          "metricSpecifications": [
            {
              "name": "ResourceUtilization",
              "displayName": "SU % Utilization",
              "displayDescription": "SU % Utilization",
              "unit": "Percent",
              "aggregationType": "Maximum",
              "supportedAggregationTypes": [
                "Average",
                "Maximum",
                "Minimum"
              ],
              "availabilities": [
                {
                  "timeGrain": "PT1M",
                  "blobDuration": "PT1H"
                },
                {
                  "timeGrain": "PT30M",
                  "blobDuration": "PT1H"
                },
                {
                  "timeGrain": "PT1H",
                  "blobDuration": "PT1H"
                }
              ],
              "dimensions": null
            },
          ]
        }
      }
    },
    {
      "name": "Microsoft.StreamAnalytics/clusters/Delete",
      "display": {
        "provider": "Microsoft Azure Stream Analytics",
        "resource": "Cluster",
        "operation": "Delete Cluster",
        "description": "Delete Cluster"
      }
    },
    .
    .
  ]
}
```

Update Subscription State API

The ARM RPC dictates what we should when we are notified that a subscription moves from a Registered state to a different state (see table below taken from the ARM RPC). For example, when a subscription goes into a Warned states, it indicates to us this subscription no longer has create/update permissions (i.e. PUT/PATCH/POST) on our resources and can only retrieve or delete their resources (GET/DELETE).

Currently this API only handles streaming jobs, but now we will need to extend it to also handle clusters and private endpoints that are within the subscription. This may mean turning off the VMs (not deleting them) when a subscription goes into Warned state and then turning them back on if the subscription goes back to Registered state, for example.

SubscriptionState	Description
Registered	The subscription was entitled to use your "ResourceProviderNamespace". Azure will use this subscription in future communications. You may also do any initial state setup as a result of this notification type. When a subscription is "fixed" / restored from being suspended, it will return to the "Registered" state. All management APIs must function (PUT/PATCH/DELETE/POST/GET), all resources must run normally; Bill normally.
Warned	The subscription has been warned (generally due to forthcoming suspension resulting from fraud or non-payment). Resources must be offline but in running (or quickly recoverable state). Do **not** deallocate resources. GET/DELETE management APIs must function; PUT/PATCH/POST must not. **Don't emit any usage. Any emitted usage will be ignored.**

Suspended	The subscription has been suspended (generally due to fraud or non-payment) and the Resource Provider should stop the subscription from generating any additional usage. Pay-for-use resource should have access rights revoked when the subscription is disabled. In such cases the Resource Provider should also mark the Resource State as "Suspended." We recommend that you treat this as a soft-delete so as to get appropriate customer attention. GET/DELETE management APIs must function; PUT/PATCH/POST must not. **Don't emit any usage. Any emitted usage will be ignored.**
Deleted	The customer has cancelled their Windows Azure subscription and its content *must* be cleaned up by the resource provider.The resource provider does *not* receive a DELETE call for each resource – this is expected to be a "cascade" deletion.
Unregistered	Either the customer has not yet chosen to use the resource provider, or the customer has decided to stop using the Resource Provider. Only GETs are permitted. In the case of formerly registered subscriptions, all existing resources would already have been deleted by the customer explicitly.

References

1. Azure Resource Manager Resource Provider Contract (ARM RPC)
2. ARM Wiki
3. Azure Private Endpoint Overview
4. Azure Private Endpoint REST API Documentation