

PowerBI Whitelisting

Steps:

- 1) Download the IP List from Microsoft
- 2) Update the IPs in Snowflake Network Policy

1) Download the IP List from Microsoft

First, we need to create a script that can download through the internet the file with all the Services IPs Microsoft has. This must be downloaded and placed in storage accessible to Snowflake (in our case GCP blob).

Here is a sample command:

```
curl -sS https://www.microsoft.com/en-us/download/confirmation.aspx?id=56519 | egrep -o 'https://download.*?\.json' | uniq
```

Other examples can be found here:

<https://stackoverflow.com/questions/28798014/is-there-a-way-to-automatically-and-programmatically-download-the-latest-ip-rang>

2) Update the IPs in Snowflake Network Policy

Secondly, we would then ingest the file and parse through the data to pull only the IPs for the MS Service needed (in our case PBI).

Here is some code attached that will outline how we can create a stored procedure that will look for the file and then update the Network policy by passing in the SP parameter the service account needed.

SP to ingest file and create Network Policy

```
CREATE or replace FILE FORMAT MS_IP_JSON
  TYPE = 'JSON'
  COMPRESSION = 'AUTO'
  ENABLE_OCTAL = FALSE
  ALLOW_DUPLICATE = FALSE
  STRIP_OUTER_ARRAY = TRUE
  STRIP_NULL_VALUES = FALSE
  IGNORE_UTF8_ERRORS = FALSE;

create or replace stage microsoft_ip_stage
  file_format = MS_IP_JSON;
create or replace procedure update_azure_network_policies
(SERVICETAG String, FILENAME STRING)
returns string not null
language javascript
  Execute as owner as
$$

  var sql_to_clean_json_file = `CREATE OR REPLACE TABLE AZUREDB(src variant) as select value as src from table
(flatten(select parse_json($1):values from @HRDP_LND_DV_DB.LRN_LND_SCH.GCS_INT_STAGE_DV/successfactors
/POWER_BI_IP_LIST/azure_powerbi_ips.json (file_format => 'MS_IP_JSON')));`;

  var statement1 = snowflake.createStatement( {sqlText:sql_to_clean_json_file} );
```

```

var result_set1 = statement1.execute();
    // Uncomment the following line to see the generated statement.
//return sql_to_clean_json_file;
// Step 2. Transform the result into a table. Extract the IP address
// array into a string of comma-separated IP addresses for use in a Network
// Policy.
var sql_to_extract_ips_and_service_tags = "";
var sql_to_extract_ips_and_service_tags = `create or replace table TAGS_IPS_RAGES
(SERVICE_TAGS string,REGIONS string,IP_Prefixes String)
as select src:id::string as
    SERVICE_TAGS,src:region::string as REGIONS,concat
        ('\\',
        array_to_string(src:properties:addressPrefixes,
        '\\',\\'), '\\')
    as IP_Prefixes from AZUREDB;`;
var statement2 = snowflake.createStatement( {sqlText:sql_to_extract_ips_and_service_tags} );
var result_set2 = statement2.execute();
    // Uncomment the following line to see the generated statement:
//return sql_to_extract_ips_and_service_tags;
// Step 3. Ignore IPV6 and Get the list of IPv4 to be inserted into the service tag Network Policy.
var sql_to_get_list_of_service_ips = "";
var sql_to_get_list_of_service_ips = `select split_part(IP_Prefixes, '\\2603:',1) as IPS from
HRDP_ADM_DV_DB.SF_MONITOR_SCH.TAGS_IPS_RAGES where SERVICE_TAGS='` + SERVICETAG + `' limit 1;`;
var statement3 = snowflake.createStatement( {sqlText: sql_to_get_list_of_service_ips} );
var result_set3 = statement3.execute();
    var get_the_first_row = result_set3.next();
var ips_column_name = "IPS"
var quota_separated_list_of_ips = result_set3.getColumnValue(ips_column_name);
    // Uncomment the following line to see the generated statement:
//return quota_separated_list_of_ips;
// Step 4. Create the Network Policy for that service tag without activating it.
// Only activate the Network Policy after verifying it!
var sql_to_create_network_policy = "";
var sql_to_create_network_policy = `create network policy
    azure_network_policy_for_` + SERVICETAG + `
    ALLOWED_IP_LIST =
    (` + quota_separated_list_of_ips + `);`
var statement4 = snowflake.createStatement( {sqlText:sql_to_create_network_policy});
try {
    statement4.execute();
    return sql_to_create_network_policy;
}
catch (err)
{
    var sql_to_alter_network_policy = `alter network policy
        azure_network_policy_for_` + SERVICETAG + `
        set    ALLOWED_IP_LIST =
        (` + quota_separated_list_of_ips + `);`
    var statement5 = snowflake.createStatement( {sqlText:sql_to_alter_network_policy});
    statement5.execute();
    return sql_to_alter_network_policy;
}

$$;

call update_azure_network_policies('PowerBI','azure_powerbi_ips.json');

```