Kunal Chandratre

Cloud Solution Architect @Microsoft

Azure API Management DR without Premium SKU

Abstract

This document provides step by step documentation on configuration Azure API Management Disaster Recovery in any pricing sku other than premium.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Updates | Date | Owner |
| 0.1 | Initial Draft | 26th Mar 2019 | Kunal Chandratre |
| 0.2 |  |  |  |

# Version History

# Azure API Management Disaster Recovery in Standard – the need?

API Management (APIM) helps organizations publish APIs to external, partner, and internal developers to unlock the potential of their data and services. Businesses everywhere are looking to extend their operations as a digital platform, creating new channels, finding new customers and driving deeper engagement with existing ones. API Management provides the core competencies to ensure a successful API program through developer engagement, business insights, analytics, security, and protection. You can use Azure API Management to take any backend and launch a full-fledged API program based on it.

Azure API Management as of today offered in various pricing sku with various features –

1. Consumption
2. Developer
3. Basic
4. Standard
5. Premium.

When you onboard API M as a part of your API story for all of your Enterprise application; this component becomes mission critical. As a best practice we should also configure Disaster Recovery (DR – multi region deployment) for API Management.

API M DR is by default provided as a service in Premium sku. However at the same time the capacity or workload support by premium sku is massive. Therefore premium sku also comes with premium price. Most of the enterprise needs are observed to get satisfied by BASIC or STANDARD pricing tier. As of today as a benchmarking API M with STANDARD tier support approx. 2500 req/ seconds per unit. STANDARD tier can have upto 4 units. Means total number of requests that can be served by STANDARD API M can go upto 10000 req/ sec. This is massive scale and satisfies most of the Enterprise requirements. **However STANDARD sku do not provide DR pre-configured.** This proposes bigger challenge for organizations who are betting big on API M.

This document provides step by step document for configuration of DR for API M STANDARD sku along with architecture. Hope this document will help to address the concern of API M DR in Standard mode.

# Solution Overview

As a part of the solution we will use below services in Azure to implement API M DR in standard mode.

1. 2 Logic apps – for automation of backup from primary region and restore in secondary region of API M
2. API M in two regions – Central India(Primary - CI) and South India(Secondary - SI)
3. Traffic Manager – to distribute the load in case of DR situation
4. Storage account – for storing backup information of API M primary
5. Azure AD application – to allow logic app provide authentication mechanism to make changes in API M (through Service Principal concept)

# High Level Steps

1. Create two API Management in Standard mode. One in CI and other in SI.
2. Configure APIs and it related backend in API M primary.
3. Create Azure AD Application to have necessary permission to make changes in API M instances in cross region.
4. Allow rights on resource groups of both API M in two regions.
5. Create two logic apps. One to create backup of primary API M. Second will be used to restore the same backup in secondary region API M.
6. Create traffic manager and add API M IP address/ custom domains as endpoint profiles. Configure Traffic manager in Failover mode.
7. Configure status page of API M in Traffic manager to know health status of API M for traffic manager so that it can switch to secondary region in case of DR situation.

# Architecture



1. Logic app uses timer orchestration step to et automatically trigger. As a next step same logic app sends service principal details to Azure AD and then retrieves the authentication token.
2. Invoke API M management API with operation as backup.
3. The operation starts copying data to storage account in other region. Depending on the size and number of APIs being used in API M the backup operation takes approximately 1-2 hours.
4. Post successful copy operation of backup; another logic app is triggered which restores the backup data in another API M instance created in secondary region.
5. Post restore operation both API M instances are exactly identical. For example, in secondary API M all policies, subscription key, APIs etc. will be same as primary API M.
6. The request for API is received from traffic manager. Traffic manager is configured in failover mode with primary API M as active and secondary API M as passive.
7. Traffic manager passes on the incoming request to active API M instance.
8. The health status page of both API M instances /status023456789abcdef keeps sending 200 response to traffic manager. In case if primary status page sends non-200 response then Traffic manager will route the traffic to secondary API M automatically.

# Certificate and custom domain considerations

Both API M instances gets same custom domain attached. The certificate to be procured should wild card certificate and same have to be used on both API M instances. There is no SSL cert upload to be done on traffic manager. For example, if we have configured the custom domain to API M instances as my.apim.com then procure the certificate as \*.api.com or with same host name as my.apim.com and upload to both API M instances. Both API M instances et same custom domain mapped as my.apim.com.

Also we need to map the same custom domain my.apim.com to mytm.traffic manager.net and DNS entry to be made for traffic manager domain only and not for API M default domains provided by Azure. So your DNS entry would only look like below –

My.apim.com IN CNAME mytm.trafficmanager.net

Also traffic manager endpoint will be default URLs of API M instances and to be added as external endpoint in traffic manager.

# Step by step execution

Here onwards we will describe how all of the API M DR with logic apps configuration can be achieved along with required screenshots.

## Resource group creation

Create two resource groups in Azure portal. One with name as APIMPrimaryRG in Central India (or in region of your choice) and APIMDRRG in South India (or in region of your choice).

To create resource groups refer - [https://docs.microsoft.com/en-us/azure/azure-resource- manager/manage-resource-groups-portal#create-resource-groups](https://docs.microsoft.com/en-us/azure/azure-resource-%20manager/manage-resource-groups-portal%23create-resource-groups)

## API M instance creation

Create one instance of API M name as APIMPrimary in central india resource group in standard mode and another one as APIMDR in south indiaresource group.

Refer - <https://docs.microsoft.com/en-us/azure/api-management/get-started-create-service-instance>.

## Service principal creation

Service principal will be required to authenticate logic apps so that I can perform management operation of backup and restore on both of the API M instances. For creating service principal refer the below documentation - <https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal>. Follow only till create step.

Then follow the guide <https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#get-values-for-signing-in> to get authentication key and application id. Record it for future use.

## Assign delegated permission to Service principal

Refer document - <https://github.com/Azure/api-management-samples/tree/master/tutorials/automating-apim-backup-restore-with-logic-apps>. Out of this only follow the document section related to “Assign delegation permissions” and “Assign resource group roles”. This provides all the necessary permissions for service principal and ultimately to logic apps to perform management operations on API M instances.

## Create Storage account

Create storage account named as apimdrstor as per the steps here in Dr region resource group - <https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal>.

Then create container named as “apimbackup” in this storage account.

## Logic app – Backup operation for API M

Create Logic app as described o the link. Make sure that you name the Logic App as “APIM\_Backup\_LogicApp”. <https://docs.microsoft.com/en-us/azure/logic-apps/quickstart-create-first-logic-app-workflow#create-your-logic-app>

Follow only Create step of Logic app and then click on “Blank Logic App”. Do not follow rest of the article.

Now click on “Logic App code view” option and then copy paste the code of file “APIM-Backup-LogicApp-CodeView – Generic.json” in this repository.

Also replace the value of below setting with you own values –

"accessKey": "StorageAccountAccessKey-you created in Dr region resource group",

"apimInstance": "name of primary region API M instance",

"applicationId": "Service principal application ID",

"backupName": "Name of the backup file.",

"clientSecret": "Service principal secret Key",

"containerName": "name of the container you added in storage account.",

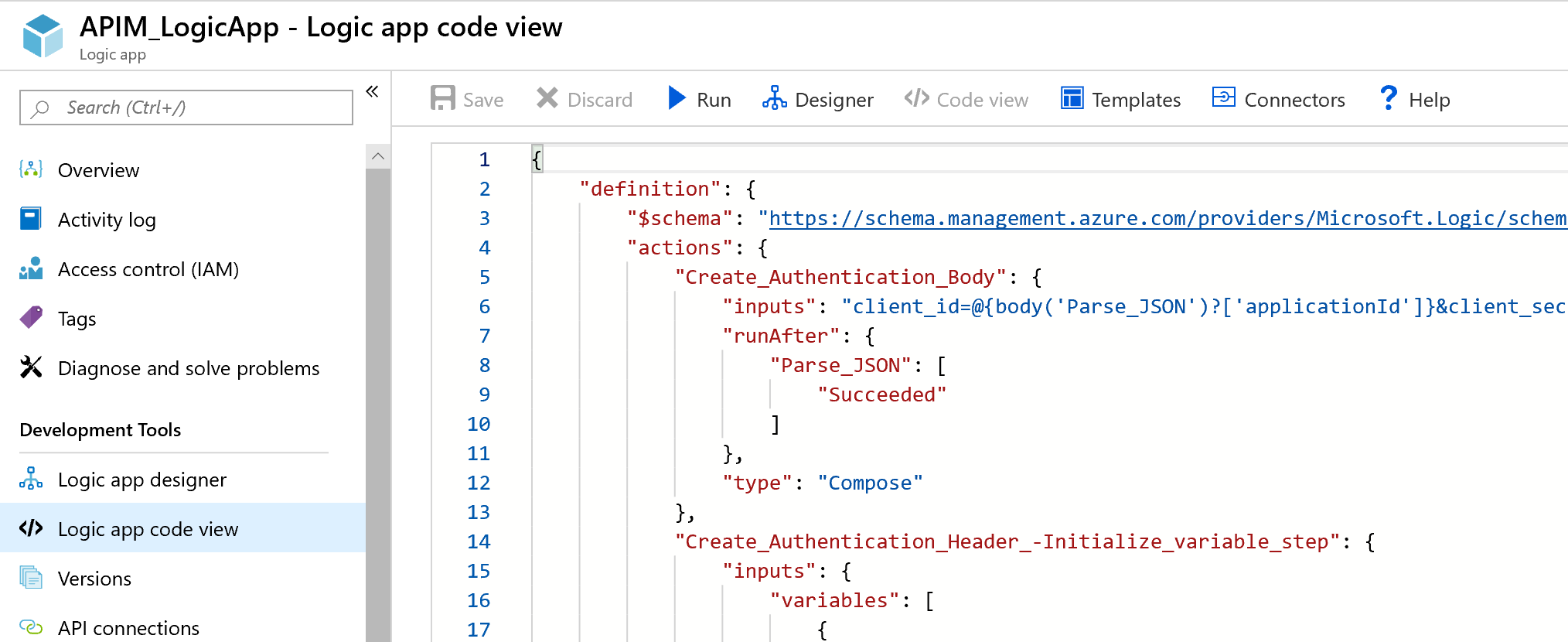
"operation": "backup", - **don’t make any change here.**

"resourceGroup": "Resource group name of primary region where primary API M instance is present",

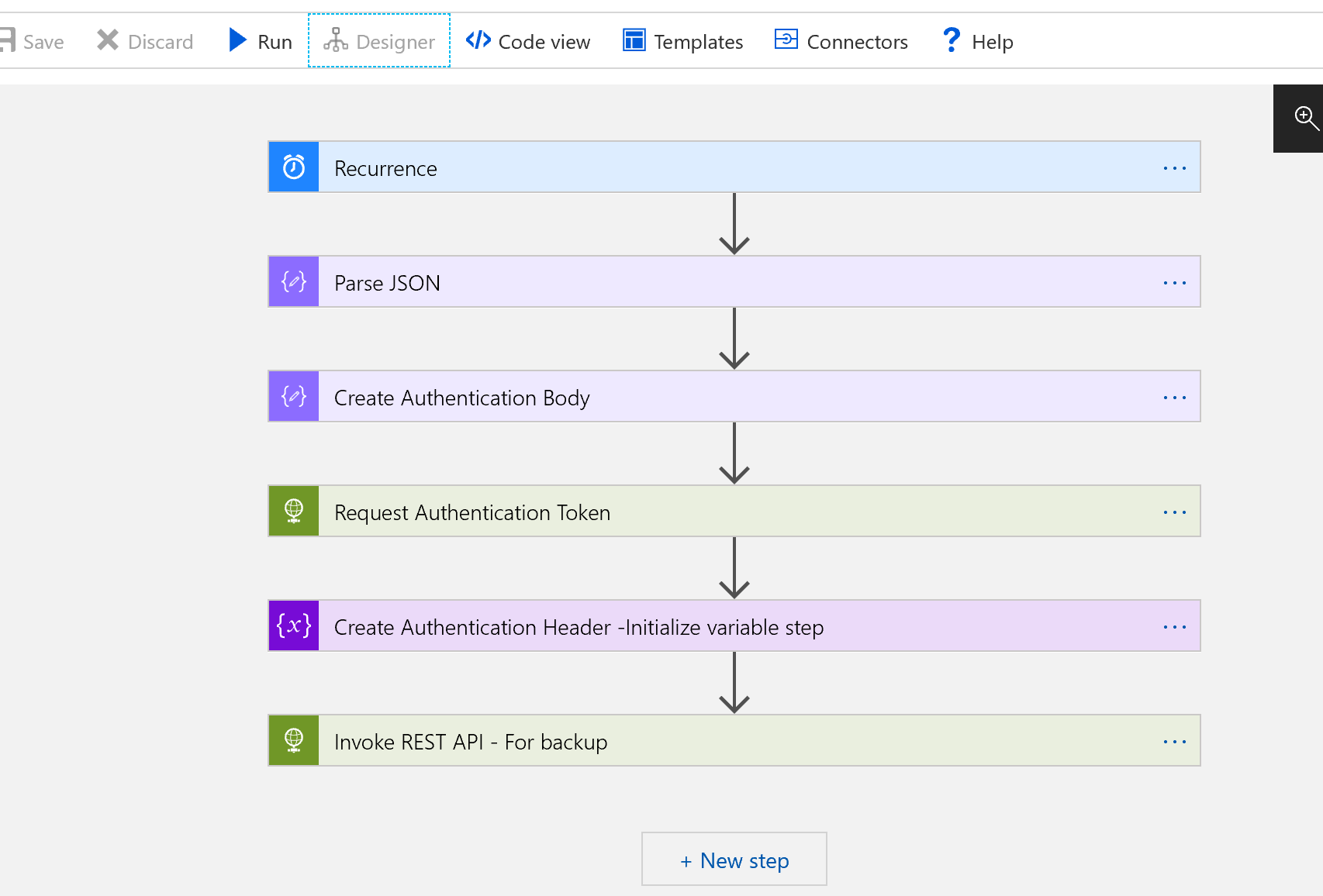
"storageAccount": "name of the storage account created in DR region resource group",

"subscriptionId": "Your Azure subscirpiotn ID",

"tenantId": "Your Azure AD tenant ID"



Now open the designer view. Steps created will be as shown –



Configure the Recurrent step as per your requirement. As a guideline configure the recurrence per day during night time/ off business hours.

This completes the configuration backup Logic app. Enable and it should execute automatically based on the schedule.

## Logic app – Restore operation for API M

Create new Logic app in DR region resource group only; named as “APIM\_LogicApp\_Restore” same as previous step with Blank template and copy paste the same template code of above backup step on the code view. Make changes in the values as shown below.

accessKey": "StorageAccountAccessKey-you created in Dr region resource group",

"apimInstance": "name of secondary region API M instance",

"applicationId": "Service principal application ID",

"backupName": "Name of the backup file on which backup is created in storage account blob.",

"clientSecret": "Service principal secret Key",

"containerName": "name of the container you added in storage account.",

"operation": "restore", - **don’t make any change here.**

"resourceGroup": "Resource group name of secondary region where secondary API M instance is present",

"storageAccount": "name of the storage account created in DR region resource group",

"subscriptionId": "Your Azure subscirpiotn ID",

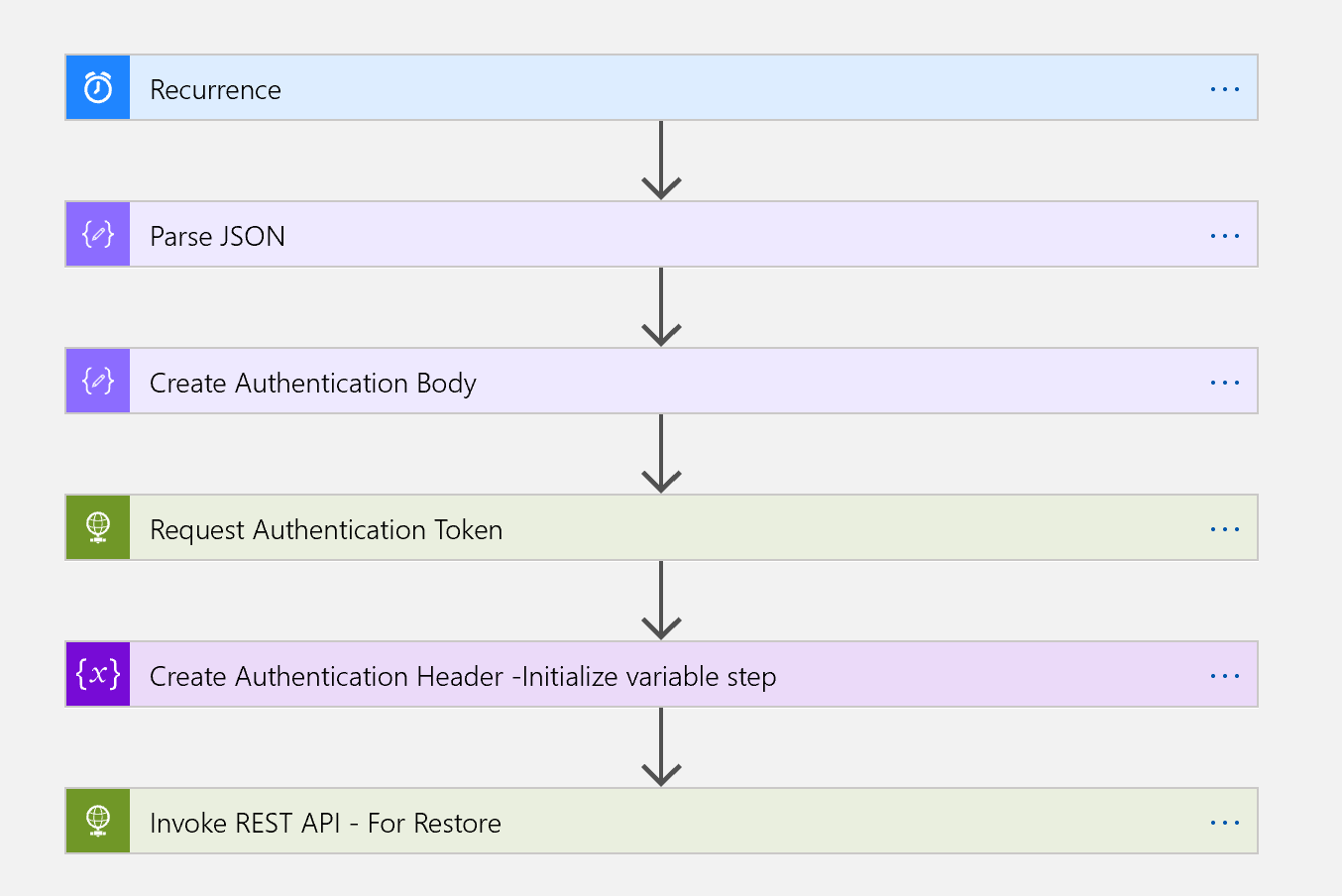
"tenantId": "Your Azure AD tenant ID"

Also make sure that you change the name/ heading of last operation step to Restore instead of backup.

Make sure you select DR region resource group. Click on purchase to complete the deployment.

This deploys the logic app required for backup operation of API M. Now we need to follow additional steps.

Your logic app for restore process will be show as below in designer view –



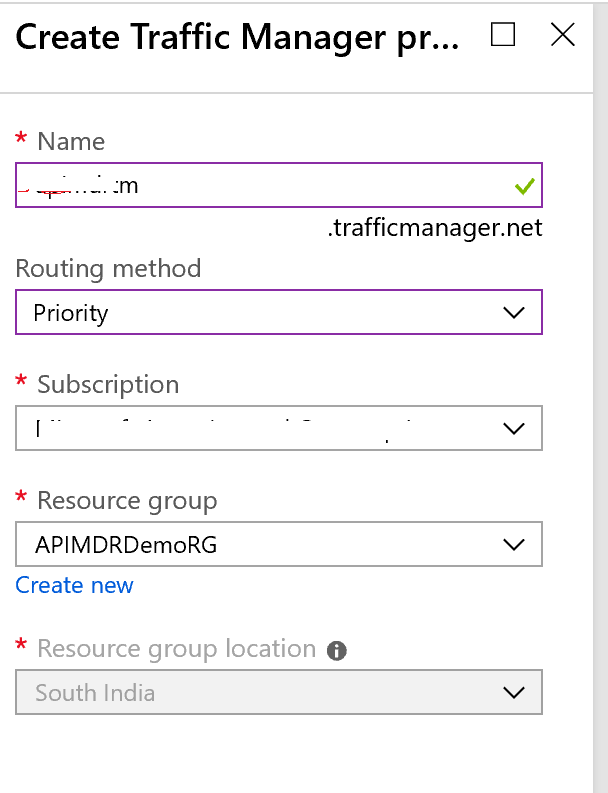
Change recurrence steps as per your need. General best practice is you should have 2 hours of difference between backup and restore operation. For example you are starting backup at 11PM then start restore at 1AM.

The above steps completes/ syncs to API M instances. No we need to configure traffic manager for distributing load to primary or secondary in case of disaster.

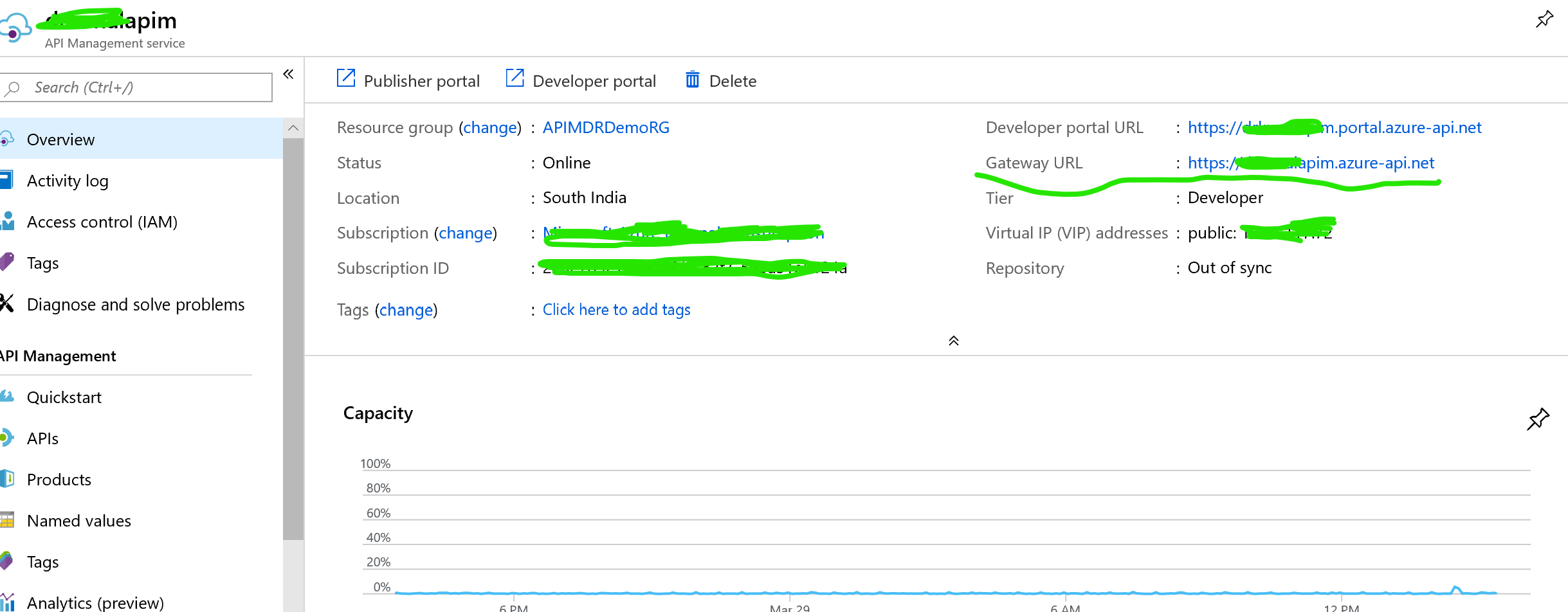
## Traffic Manager configuration

Create traffic manager as per the link described in secondary region resource group – <https://docs.microsoft.com/en-us/azure/traffic-manager/quickstart-create-traffic-manager-profile#create-a-traffic-manager-profile> .

Provide values as shown below –

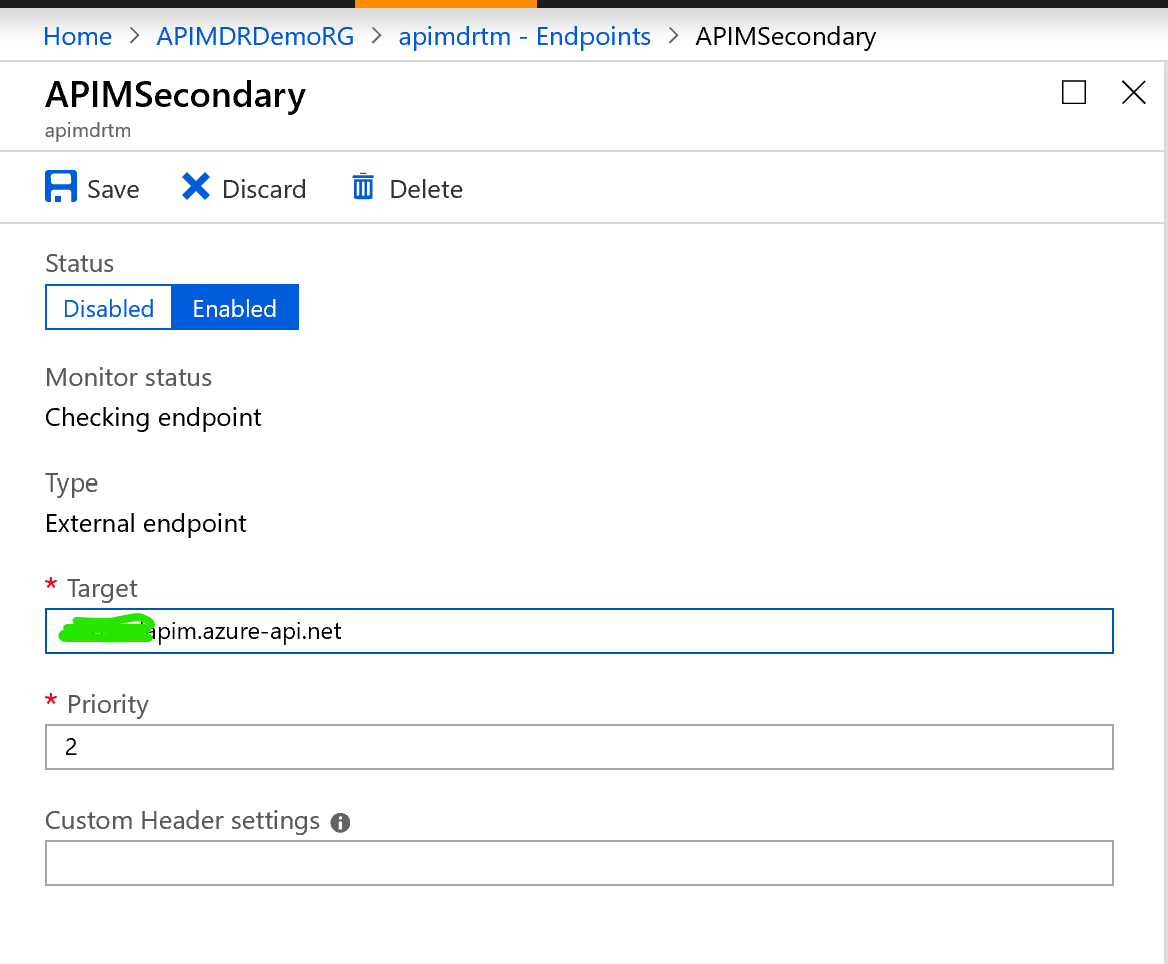


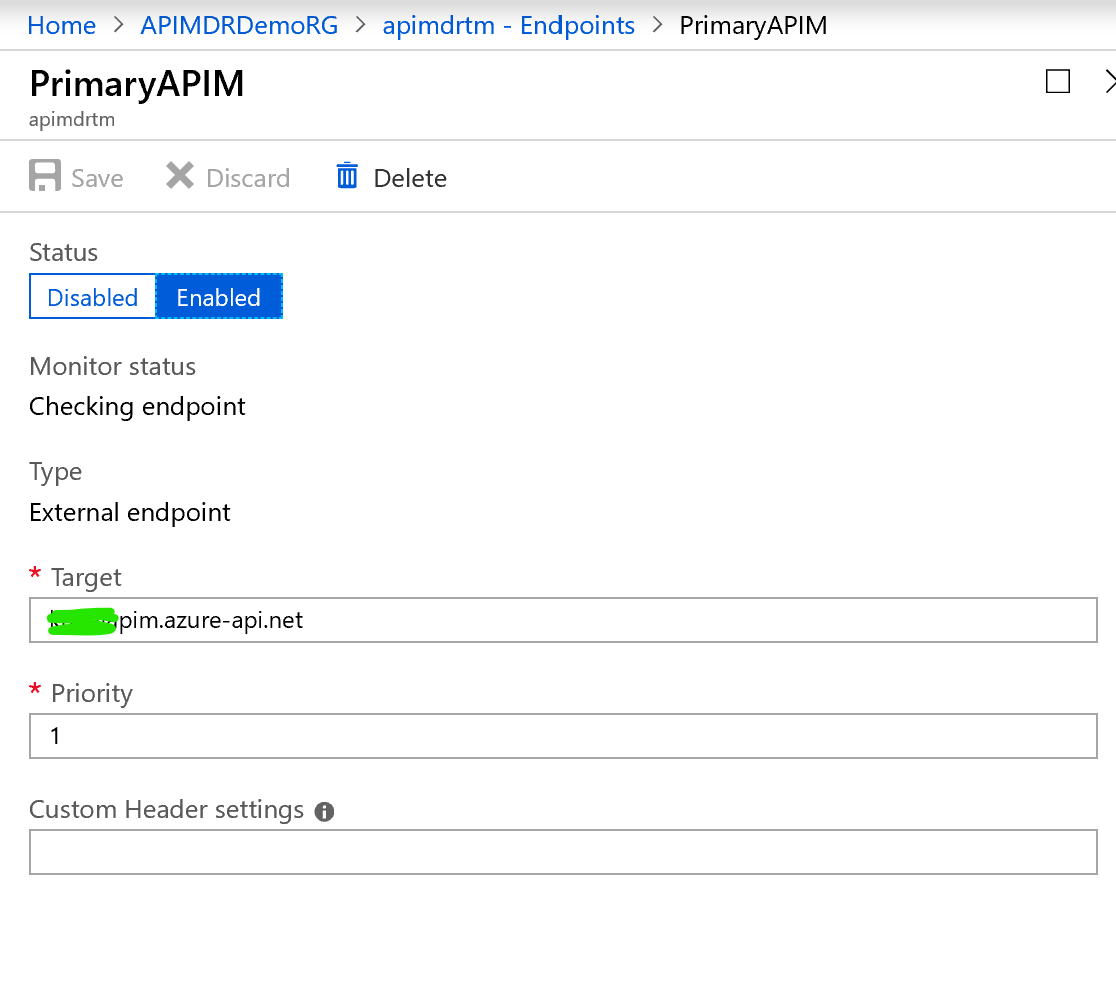
Not the IP addresses of API M standard instances from primary and secondary instances as shown below –



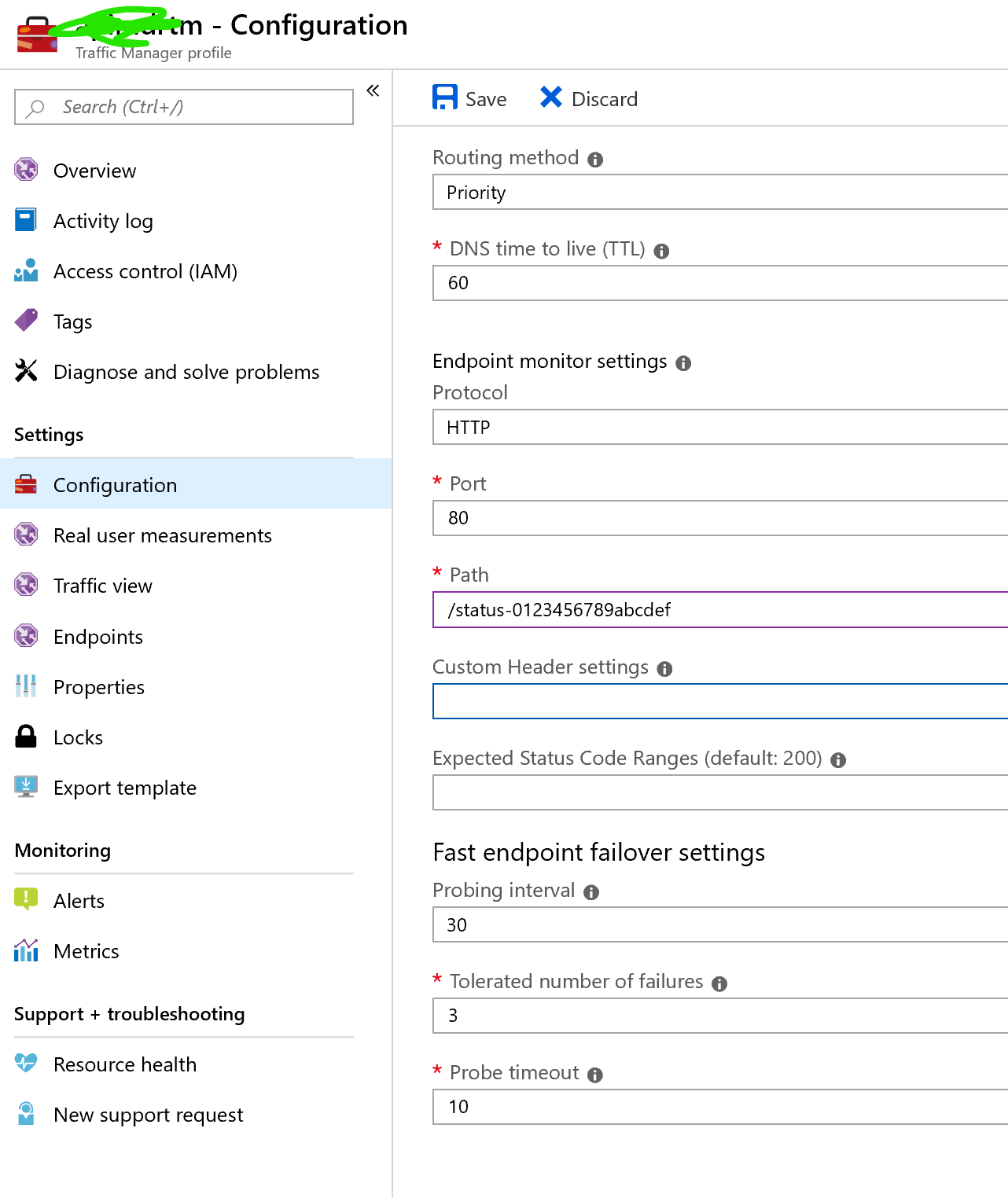
We will use these IP addresses as an endpoint in traffic manager profile.

Configure the traffic manager endpoint as shown below –





Configure the health status page for API M –



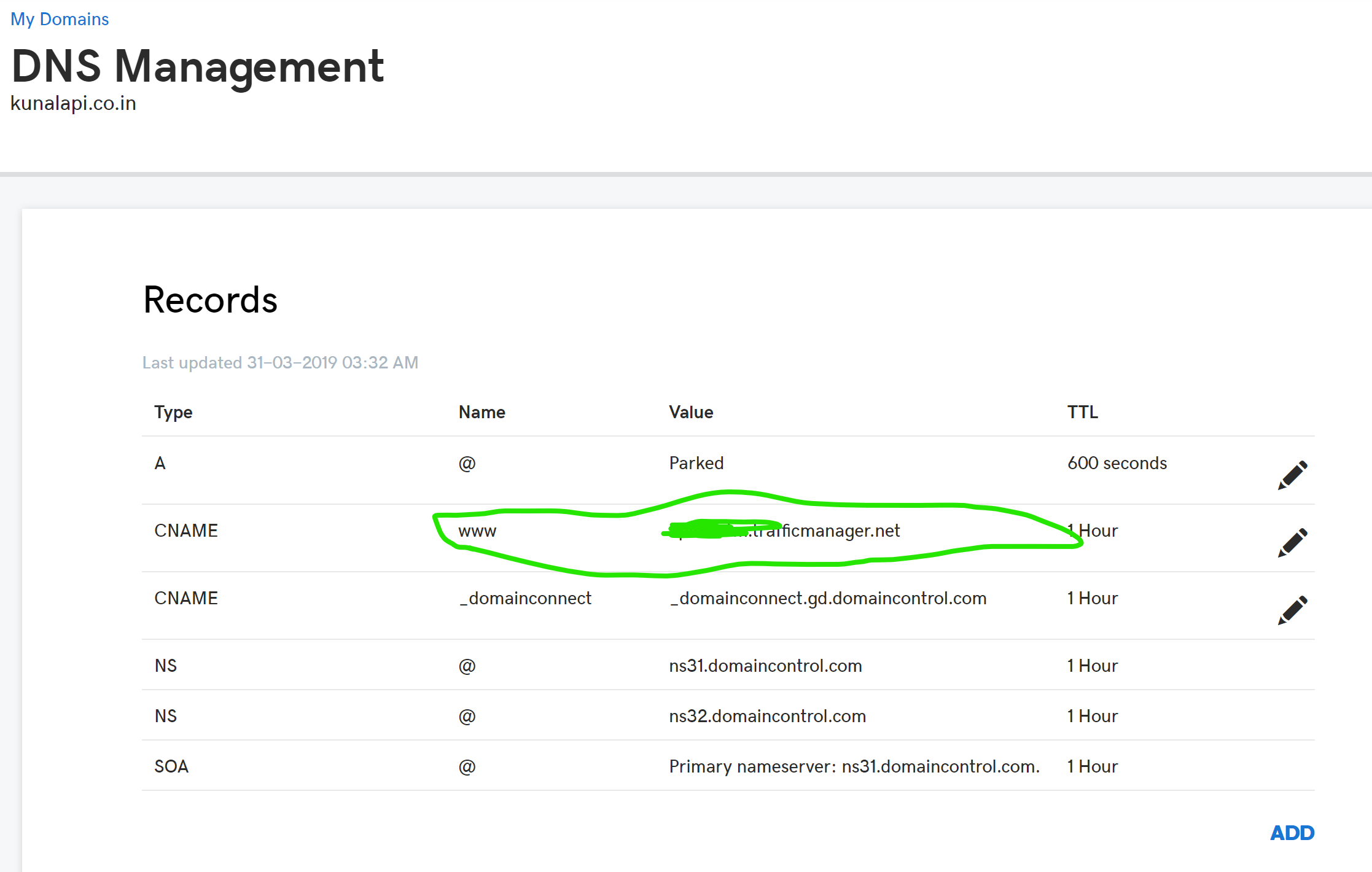
The port number above will change to 443 and protocol will be https in case you are planning for https configuration.

## Custom Domains and certificate configurations

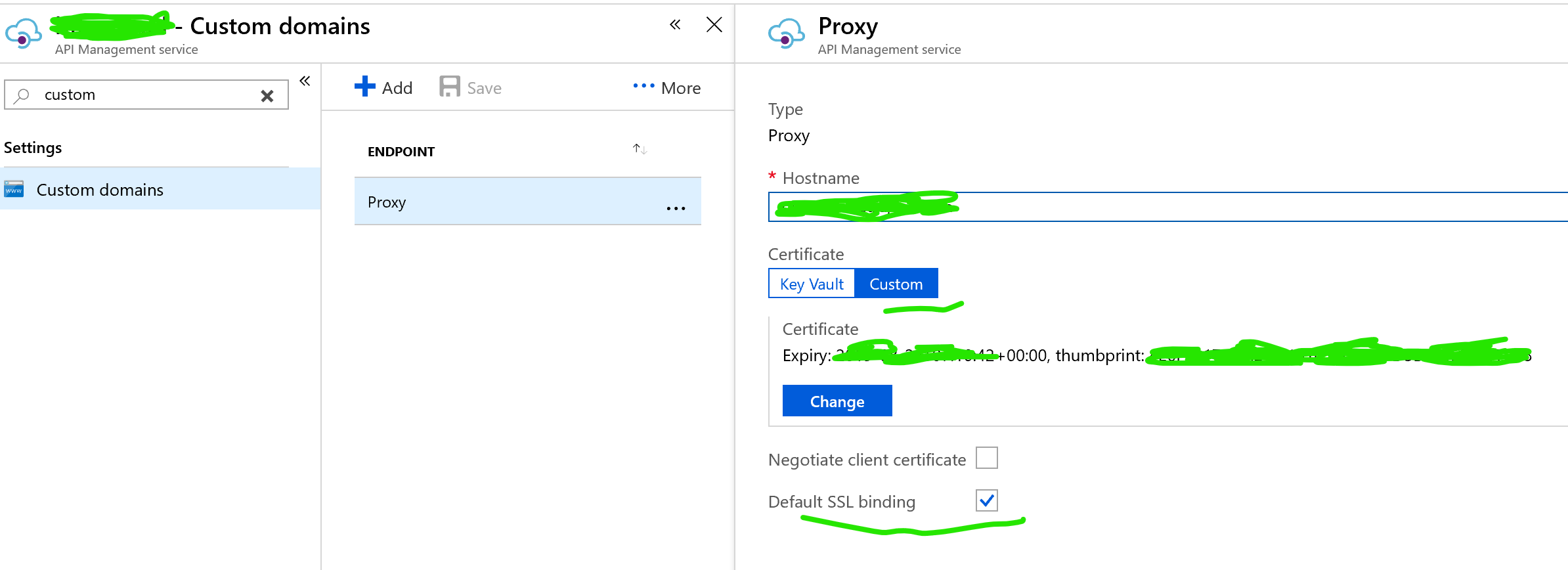
Point your CNAME to traffic manager URL

my.api.com IN CNAME mytm.trafficmanager.net

(for example – GoDaddy screenshot )–



Configure the same custom domain at both API M. Also, if you are using certificates then make sure they are mapped to my.api.com (-your custom domain name) or wild card certificate like \*.api.com. For testing purpose you can use the self signed certificate same as your custom domain and it should work for testing purpose.

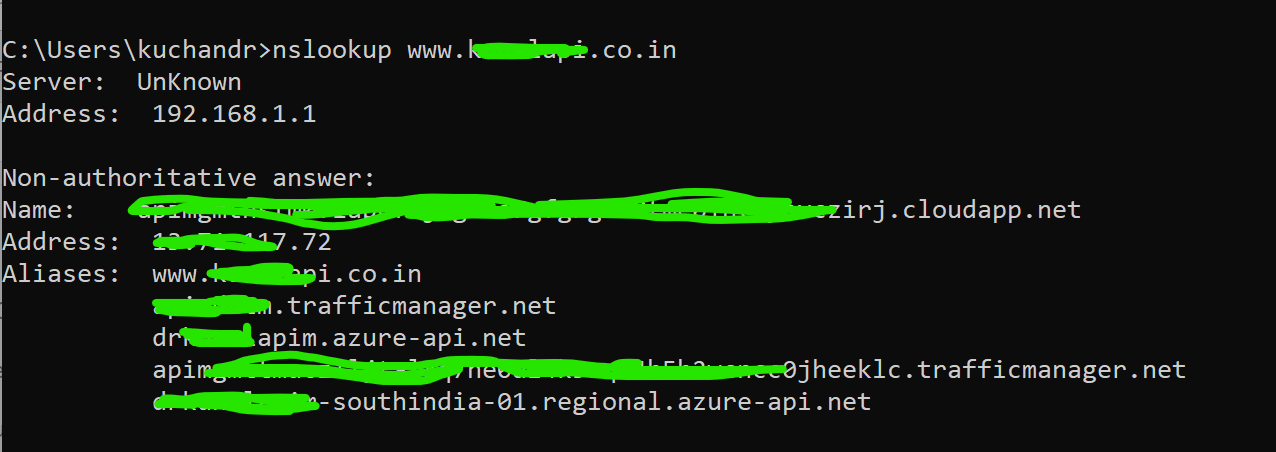


## Testing the Scenarios

You cannot customize the status of API M to respond non 200 response. In case of DR situation only status page will respond non 200 response.

Therefore for testing purpose configure an API Page URL as a part of health page in above screenshot under “Path” value; use APIM Policy to mock response of the API to non-200 and check if Traffic Manager is sending request to secondary API M instance.

Similarly to make sure that traffic is flowing through primary or DR region fro traffic manager you can use nslookup to your custom domain. Example below –



# Troubleshooting reference

Traffic manager sending 503 response.

<https://social.msdn.microsoft.com/Forums/azure/en-US/4f44613b-8c0e-4482-b1c7-4703c7fc979c/api-management-and-traffic-manager-503-service-unavailable?forum=azureapimgmt>