Who needs multicloud IP address management and DNS orchestration?

Where do clouds start and stop

Both humans and machines use names and labels to find and access resources. When crossing from the physical to the digital realm, each element or entity requires a label or address for it to be accessed or used. Internal and external networks have boundaries. They are architected using a range of topologies and then grouped depending upon their characteristics, function, and risk level. Profiles and policies are then applied to these network zones and enclaves to facilitate sharing or enforce segregation.

Multicloud, a relatively recent term, speaks to the extension of our IT footprints to 3rd party zones and services. To be multicloud means to depend on or build on remote clouds with layers outside of our direct governance and control. Technically, we've been doing this for a long time, depending on how far up or down the stack you choose to go. We used to call trusted or semi-trusted zones that offered reachability or access to 3rd party services, an extranet. Albeit we might not have programmatically consumed cloud infrastructure services, moreso application services, we nonetheless consumed remote services from remote clouds.

Once an organization consumes services from any external cloud outside of its own fully managed network, it could effectively be thought of as engaging in a form of multicloud architecture. As Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) extend our organizational footprints into new territories, there is no escaping basic Asset Management for foundational services such as DDI (DNS, DHCP, and IPAM). As footprints inevitably grow and change, multicloud IP Address Management (IPAM) and DNS administration become even more crucial for the digital assets you name and control. Each new edge, network, or cloud-based resource increases the burden of operational responsibility. Unified DDI management and orchestration become table stakes to keep the packets flowing.

The lynchpin that is DNS

"It's not DNS. There's no way it's DNS. It was DNS" is an utterance we want to avoid and particularly in relation to high-priority incidents. The impact of DNS-related incidents can span more than just an organization. They can ripple outwards affecting customers, suppliers, prospects, and users. Minimizing risk then begins with being able to see the whole picture, ensuring a single Source of Truth (SoT), and elevating DNS hygiene and Change Management practices. Now, more than ever, it's an operational necessity to have a trustworthy IPAM, one tightly integrated with a unified DNS management platform. By eliminating common mistakes and catching errors early, everything from early project design to deployments and migrations benefits. With better team collaboration and assurance, harm is minimized, confidence is bolstered, and peace of mind gets built into frictionless workflows.

Microsoft Azure DNS and Active Directory (AD)

With 21% of the public cloud market share (read more), Microsoft Azure is the second most dominant player. Buoyed by a huge and existing onpremises market, Azure's focus is not just on cloud natives and net new customers, but on a seamless journey towards cloud adoption for those already immersed in Microsoft technology. By enabling cost-effective workload migration from existing on-premises footprints, overall complexity can be minimized while change is de-risked. Additional benefits for existing Microsoft AD (Active Directory) users come in the form of rapidly being able to adopt hybrid cloud strategies, especially for identity services. By using Azure AD Connect, organizations can radically simplify with a common identity for accessing both on-premises and cloud services like Microsoft 365. Azure AD Connect provides synchronization of users, groups, and objects while also offering federation integration, health monitoring, enhanced security, and an overall improved user experience.

What drives the use of multicloud DNS?

Growth, diversity, and evolution

As businesses move forward and expand through time and space; their IT footprints morph and grow. Whether it's due to planned growth or accelerating evolution, technology marches steadily onwards, and IT resources and services can be perceived as being in a state of flux. From inception through to delivery and decommissioning, lifecycle expectations often front-run the reality on the ground. IT teams are on the hook for constantly delivering with a minimum of friction and fuss.This ongoing challenge is best met by smarter approaches to virtual asset management which underpins lifecycle management.

As functionality and requirements inevitably change over time, each service, group, and location sees their IT footprints evolve. This leads to multiple trajectories and streams of work. Collaboration becomes key, as does the need for flexible OTT (Over The Top) solutions that preserve visibility while providing unified management of core atomic assets. IT must be able to audit, allocate, and account for DDI (DNS, DHCP, and IPAM)-related virtual assets at any stage of a project, migration, or build, and not just for operational or security reasons.

IaC (Infrastructure as Code) and IaaS (Infrastructure as a Service) are also shining spotlights on the importance of timely and accurate virtual asset management, highlighting a renewed focus on DDI. From the edge to the core and across all geographies, environments are becoming faster and more ephemeral. They are harder to track and troubleshoot. More intelligent, trusted, and automated workflows are needed. With greater assurance comes peace of mind and optimal outcomes, but only when the foundations are rock solid.

Mergers and acquisitions (M&A)

Integrating another organization's DNS and IPAM footprint is a huge undertaking. There is often little time to gain crucial visibility for subsequent management and orchestration. OTT (Over The Top) solutions excel at this challenge and ensure minimal disruption with maximum positive impact. Even if the acquiree's designs and namespaces are modular, IP spaces often include duplicate IP prefixes, which can also include disparate and heterogeneous cloud services. Acquisitions require rapid and continuous observability of core digital assets. Multicloud data management then becomes a reality, often including multiple new Microsoft Azure subscriptions and accounts.

Azure as an inevitable outgrowth

While racing to embrace cloud services, it's worth remembering that many organizations' digital footprints still exist across a wide range of locations and depend on a broad suite of vendors. Diverse digital and physical assets may live anywhere from the access edge to deep within a data center network. Workloads can run on edge computing in small computer rooms and labs or in large distributed compute clusters located in data centers.

Public clouds target distinct problem spaces like long lead times, scarcity, and inelasticity. Yet, when first embracing these greenfield clouds, blue-sky thinking may lead, but inevitably, each cloud risks becoming the brownfield of' tomorrow. Traditional management fundamentals become even more pronounced with the velocity, accessibility, and complexity of cloud computing. Automation and API-driven cloud services accelerate this game, as does SASE (Secure Access Service Edge), which itself leverages elements of SD-WAN (Software-Defined WAN).

Embracing a public cloud such as Microsoft's still entails Azure DNS and Azure IPAM records having to be fresh and accurate at all times. System observability and controllability are still key for both traditional and cloud footprints. The boundaries of IP and the challenges with DNS and namespaces persist. Irrespective of how you end up connecting users and agents to on-premises or cloud-based workloads, unified DNS, DHCP, and IPAM (DDI) solutions set you up to succeed, continually and reliably.

DDI automation, orchestration, and management

Digital borders are defined by IP addresses, prefixes, and virtual policy enforcement points. IP flows initiate and terminate on interfaces using addresses resolved from records in namespaces. Tracking, allocating, and assuring ourselves that our fundamental building blocks are sound is an arduous undertaking though it should be painless and easy.

In a world of increasingly automated service provisioning and consumption, DDI operations must be timely, concise, and correct. The challenge accelerates when embracing IaC (Infrastructure as Code) and programmatic access to dynamic digital resources. We now begin to more deeply realize that DNS is one of the most, if not the most important, active assets and directories. It facilitates resource location, defines service endpoints, and uncovers paths to dependencies across public and private clouds.

Using a single authoritative UI (User Interface) and API, an OTT (over the top) and unified DDI solution enable simple integration while streamlining interactions with each silo without displacing them. This DDI solution then becomes the one trusted System of Record (SoR). When your perspective shifts to view DNS as the beating heart of a digital footprint, it's not just operations and security teams that demand effective and efficient Change Management and record management; it also heavily impacts project and product team delivery.

Supported platforms and endpoints

It's imperative that a DDI solution can meet your current needs, but it also must be able to evolve with your organization's changing needs. By selecting a DDI platform that doesn't involve a rip-andreplace methodology, you free yourself to embrace the best-inbreed DNS and DHCP solutions. Avoiding redesign, migrations, and friction is easy with an OTT (Over The Top) manage-ment and orchestration solution. This is how smart teams preserve their existing investments in infrastructure, workflows, and expertise. A new unified and trusted authoritative layer can then meet and exceed the demands of the present and even the future.

By providing support for commonly used services such as Windows DNS, BIND, Unbound, PowerDNS, Akamai Fast DNS, Azure DNS, Amazon Route 53, NS1, and Dyn, to name a few, means accelerated time-to-value without any rework. And once your footprint extends outwards from your home cloud to other cloud providers, multicloud data management becomes ever more crucial for development, staging, and production environments.

Evolving workflows and feature sets

Business and operational processes are made up of multiple workflows. Sometimes these workflows are informal and opaque, leading to a loss of visibility, accountability, and repeatability. Opportunities for innovation and automation are diminished in this darkness. On the other hand, formal workflows that have been modeled and mapped give rise to greater consistency, faster learn-ing, and more frequent optimization. With DNS, DHCP, and IPAM tasks, there is an implicit need for accountability and traceability to give confidence and assurance throughout each and every operating environment.

With an OTT (Over The Top) DDI solution, independent best-ofbreed services can evolve at different rates at the service edge, while the core DDI platform and supported data types can evolve more steadily in an agnostic and operator productivity-focused manner. An OTT DDI solution has the luxury of focusing on addvalue macro features and workflows (rather than getting stuck in the weeds). This OTT model also gives rise to innovative new features like xDNS for cross-provider zone redundancy, seamless migrations, and superior resilience.

With innovative features and standardized workflows, a singlelayer API can now enable a fully programmable environment and unlock automation that becomes a true force multiplier. Workflows can reach beyond a single system or silo and integrate with productivity, notification, and ticketing systems. Security and operations teams can automate incident response and enrichment using a single API that covers all an organization 's DDI assets.

Azure and the Micetro REST API

In addition to Micetro's powerful UI, its fully-featured API also facilitates Azure DNS and Virtual Network actions. The Micetro API acts as a single broker to unlock and enable greater extensibility and automation across home and remote clouds. By providing a single-layer API that abstracts DDI tasks across multiple providers, teams are empowered to enhance their own workflows, build productivity-enhancing integrations, and create custom solutions.

The downstream Azure complexity and glue are fully abstracted away and common tasks are then initiated from Micetro as a single authoritative source. This enables local and remote teams to be more efficient, make smarter decisions, and collaborate across project or team boundaries. APIs and automation open up a world of previously unrealized innovation.

Once you install the free trial, there is a full set of OpenAPI (Swagger) documentation available directly on your Men&Mice web application accessed via the path **/mmws/api/doc/**, and the latest product guides are always available at https://menandmice.com/docs.

Multicloud data management and integration with Azure

How to integrate Micetro with Azure

Micetro is an overlay and orchestration solution for DDI (DNS, DHCP, and IPAM) environments, including on-premises and cloud-based assets. Two of Microsoft Azure's pivotal services are Azure DNS and Virtual Networks, and with Micetro you can take advantage of a unified and consolidated System of Record (SoR) that encompasses all your DNS footprints (and their associated Sources of Truth (SoT)).

Micetro also provides workflows, reporting, and a fully-featured API layer. This simplifies all DNS, DHCP, and IPAM operations within a unified platform, integrating heterogeneous environments rather than replacing them. Micetro uses an OTT (Over The Top) architecture to minimize upheaval, maximize efficiency, and reduce stress. To begin, we will focus on ensuring all the right elements are in place for a Micetro installation to talk to Azure (including any configuration items required on either end). This will facilitate:

- administration of zones and zone records
- administration of Virtual Networks (and their characteristics)
- (automated) asset and record synchronization
- workflows (with built-in Change Management)

Note: We will be using a subdomain called "azure" in our demo domain, "menandmice.cloud". This will allow us to clearly partition our assets and records for the reader, as other guides will focus on Amazon Web Services (AWS), Akamai Fast DNS, Openstack, etc. Your organization's specific namespace and DNS architecture will look different, of course. Micetro can integrate with and administer a whole range of scenarios and architectures. If you do require any help, have any questions, or would like some training, please reach out to us here.

Download: The following technical sections used version 10.2.2 of the Micetro suite. A fully functioning version is available as a free trial and does not require payment details.

Ensuring Micetro is ready for cloud services

Micetro is a suite of software services. The most important element of that suite is called Men&Mice Central. Men&Mice Central is the heart and soul of Micetro. All the other elements leverage it or connect to it. The following four elements will facilitate integration with Microsoft Azure (and use Micetro's Cloud Services component):

1. Men&Mice Central is required first and foremost (as mentioned). It runs on Windows or Linux and is the main hub of Micetro. It performs roles such as Identity and Access Management, database, and API access, among others. Running Men&Mice Central on Windows will provide additional insight into on-premises AD Sites and Subnets.

2. Men&Mice Web Application is the primary UI (User Interface). It is common practice to run it on the same host as Central (but this is not required) though it does require a web server like Apache or IIS to be running on the same host. It also runs on Windows or Linux.
3. Men&Mice Console is required for the initial Micetro setup (after software installation), some system administration, and certain features. All functionality is currently being integrated into the Web Application. Console only runs on Windows.
4. A Men&Mice Server Controller is required to help broker the connection to Azure. These DNS Server Controllers co-locate with BIND, Active Directory, or other DNS servers such as PowerDNS on the relevant underlying operating system

Note: We will not be walking through the basic installation of each element (1-4) above, but rather how to connect the Cloud Services component to Microsoft Azure and administer assets using it. If you need any additional installation help, the detailed walk-through guides are available here. We will link to more detailed documentation where applicable, but https://menandmice.com/docs will always point to the latest version. Please pay particular attention to the required UDP and TCP port communication between elements.

Note II: We also need to ensure that the correct licenses are installed for the DNS and IPAM modules. Additional licenses (all available and provided with the free trial) are required for the advanced Workflow and Reporting modules:

Microsoft Azure access prerequisites and current limitations

Once integrated, Micetro will import all zones and records from Microsoft Azure DNS (and also be able to create and edit them). If additionally configured for Virtual Networks integration, Micetro will also learn about Virtual Networks and their subnets, their usage, and their characteristics. It will also populate the Micetro IPAM with this data. To achieve this, Micetro requires access to Microsoft Azure using a set of Application and Object IDs (effectively used as API keys for authentication and authorization). There are detailed instructions for all cloud integrations here, but a brief Azure-specific summary follows below.

Note: The primary Micetro Central host running a Micetro DNS Server Controller must be able to speak outbound to Microsoft Azure using the port TCP 443 and be able to resolve and reach the following

URLs:

- https://management.azure.com/
- https://login.microsoftonline.com
- https://management.core.windows.net

Additionally, an Azure "Service Principal" is required, one that has the following roles:

- DNZ Zone Contributor
- Network Contributor

Custom roles and permissions

For clarity we will create a custom "Men&Mice Operator" role with all the required permissions and access.

Note: Custom roles require an Azure AD Premium P1 or P2 license.

Note II: Please ensure when creating the custom role that you use the below JSON compliant syntax in the Azure Portal (as the Powershell payload syntax is different).

To create a custom role for the "Men&Mice Operator" that can perform both DNS and IPAM actions, go to your subscription's "Access Control (IAM) / Roles / Add / Add custom role" and edit the JSON tab to reflect the below (but include your own Subscription ID).

```
{
  "properties": {
    "roleName": "Men&Mice Operator",
    "description": "",
    "assignableScopes": [
      "/subscriptions/<your Subscription ID or specify a more granular scope>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.Network/dnsZones/*",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/write",
          "Microsoft.Network/virtualNetworks/delete",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Network/virtualNetworks/subnets/write",
          "Microsoft.Network/virtualNetworks/subnets/delete",
          "Microsoft.Compute/virtualMachines/read",
          "Microsoft.Compute/virtualMachines/instanceView/read",
          "Microsoft.Network/networkInterfaces/ipConfigurations/read",
          "Microsoft.Resources/subscriptions/resourceGroups/read"
        ],
        "notActions": [],
```

Once the above role has been created, you will be able to assign it to your service principal object (next step) via your subscription's "Access Control (IAM) / Role assignments".

Microsoft Azure service principal

A service principal is an identity. It is required when you have applications, hosted services, or automated tools that need to access or modify resources.

Note: To successfully create a service principal and apply a custom role, your access level in the Azure subscription must have the permissions Microsoft.Authorization/*/Write or you must be an "Owner" or "User Access Administrator". To create the service principal with the required roles we shall:

- register an application (which generates the service principal)
- Example name: "MnM-MultiCloud-App-Registration"

Note II: Ensure you also create a client secret as we will use the "Value" (rather than the secret's ID) when configuring the Micetro integration.

- add the newly created role as a role assignment to the application
- get the tenant, subscription, and application client ID / Values

Note III: As there are multiple steps and checks to the service principal creation process, Microsoft have provided a detailed walk-through available here. The outputs we require are the object IDs and the secret "Value" from the service principal's app registration.

Tip: Don't forget to apply your custom "Men&Mice Operator" role to your service principal via your subscription's "Access Control (IAM) / Role assignments".

IDs and values required for adding Azure as a cloud service

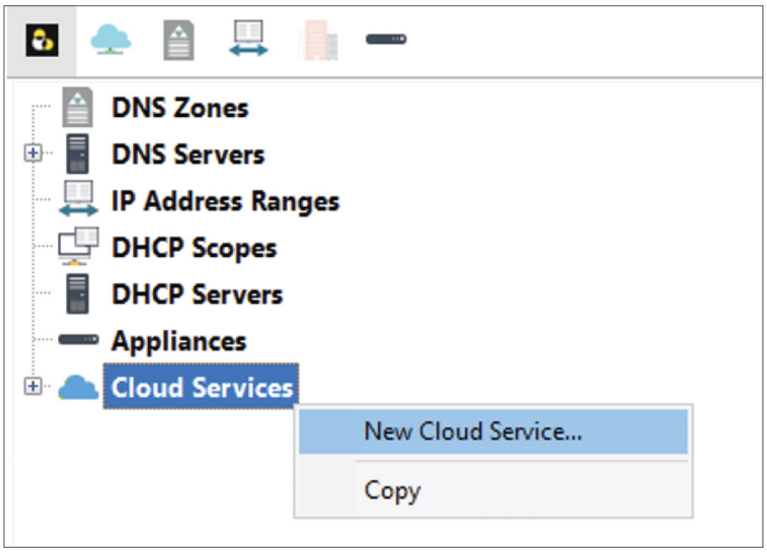
Once the above role, service principal, and role assignment steps are completed, you should now be able to retrieve your (from Azure):

- Tenant ID (from "App registrations/<your_app>/Directory (tenant) ID")
- Subscription ID (from "<your_subscription>/Subscription ID")
- Client ID (from "App registrations/<your_app>/Application (client) ID")
- Client Secret (from "App registrations/<your_app>/Client credentials/Client secrets/Value")

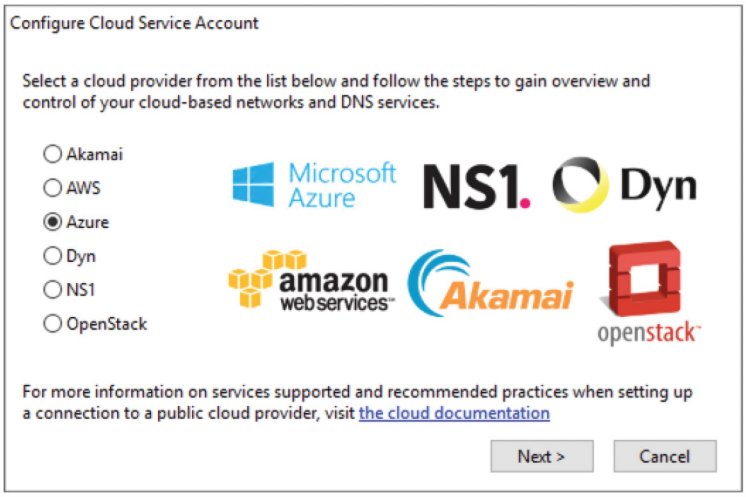
Note: Reminder that it is the secret's "Value" that is required when adding Azure as a new Cloud Service in the Men&Mice Console.

How to add Microsoft Azure as a cloud service in the console

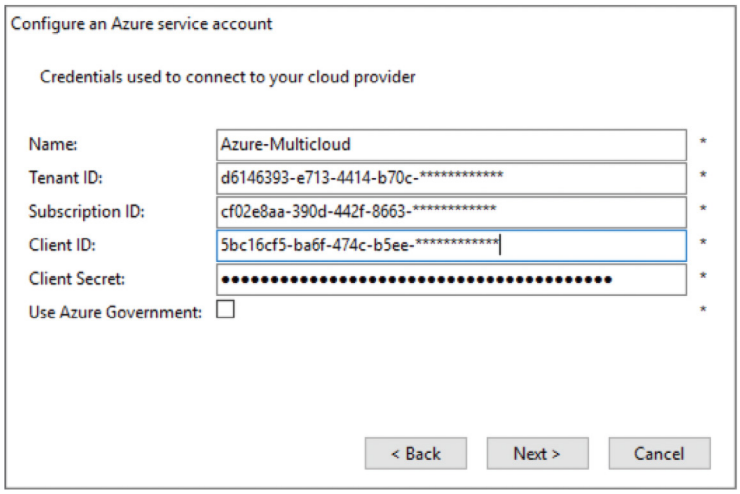
In the Men&Mice Console, go to "Cloud Services", and with a right-click, select "New Cloud Service" (or once "Cloud Services" is highlighted use the green plus symbol).



Here we can then select the type of cloud service required:

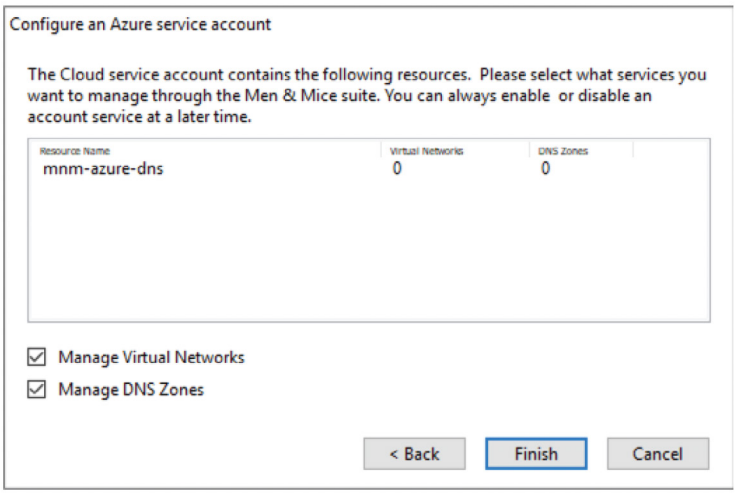


Once Azure is selected, we then want to Name our connection, enter a specific Tenant ID, its specific Subscription ID, and then both the Client ID and Client Secret (Value). The configured tenant ID (from the "App registration") must have the correct role assignment and requisite permissions for the Subscription ID (as previously mentioned)



Note: The "Client Secret" entered above in Figure 3.0 is the Secret "Value" and not the Secret ID.

Micetro will retrieve the data from the cloud provider, save the account information, and then you must select the services required.



Micetro will then synchronize with Microsoft Azure (every 900 seconds) and perform actions on demand. We are now ready for a range of common tasks and workflows including the use of Micetro's single-layer API to drive automation across our whole DNS namespace and IP footprint.

Note: All sensitive data required for communication with Azure such as IDs and secret values are encrypted both at rest and in transit.

Common tasks for Azure Cloud DNS integration

Simple secure operations

Using Micetro, a unified OTT (Over The Top) orchestration and management platform, all DDI (DNS, DHCP, IPAM) administration tasks, related access, and visibility across an organization can be centralized and controlled. Roles and responsibilities can be allocated across departments, teams, individuals, or technologies using groups and roles. Implementing a "least access principle" is easy, and it protects assets and asset classes from unauthorized changes that may lead to resource exhaustion or unexpected outages.

With management and orchestration performed by Micetro, each individual tier or entity, such as Route53, BIND, or Microsoft AD, continue to provide their own services. This architecture ensures resilience across the range of services your team already relies upon and knows well. By making everything simpler to manage while increasing visibility, less issues and errors are encountered, while better outcomes are enjoyed, more quickly.

Access management and RBAC

Within Micetro, Role-Based Access Control (RBAC) can be applied to more than just macro-level services like DNS, DHCP, IPAM, or reporting. RBAC is also applicable to specific and individual assets managed by Micetro. This allows for exceptionally fine-grained controls at an individual network, container, or zone level (if so desired).

You can use default out-of-the-box groups and roles, build on them, or start from scratch to create custom roles. You can be coarse with some groups or extremely pedantic with others. There's also a range of primitives including but not limited to; create, add, read, list, edit, delete, use, enable, and release that can be applied to objects (and their sub-objects) to create roles to govern:

- DNS servers
- DHCP servers
- DNS zones
- Ranges and DHCP scopes
- Address spaces
- Cloud networks
- Cloud services
- ISC DHCP groups

Additionally, roles for Micetro's own general administration and access can be configured and applied if the extensive range of default groups do not suffice.

Optional workflows for change management

Micetro also provides an optional Workflow module that streamlines DNS Change Management. It can be used for the creation, modification, and deletion of Resource Records (RRs). Any Micetro user (or group) given the Requester (built-in) role, such as project teams, helpdesk technicians, or other engineers, can then raise DNS change requests. The requests require approval from an administrator with the appropriate Approver (built-in) role. Once approved, changes are automatically implemented.

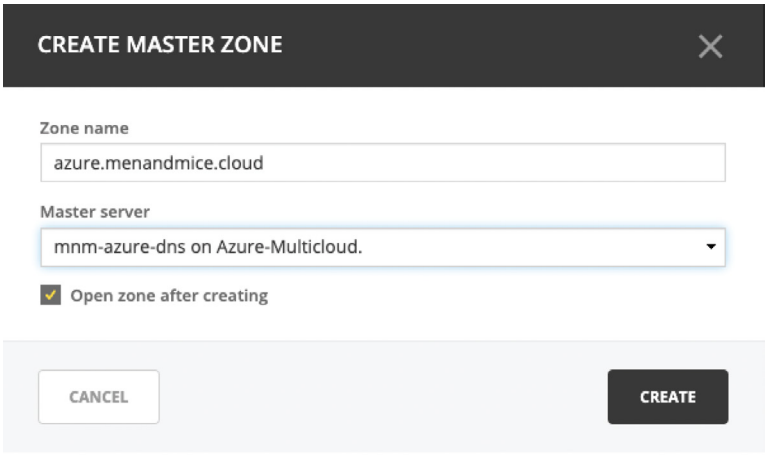
Open requests can be easily viewed and amended before being submitted for approval. Requests move through basic states such as Open, Pending, Failed, or Closed (where Closed may mean Scheduled, Fulfilled, or Rejected). Changes can even be scheduled during the request creation stage. Once approved, they are then automatically applied at the requested time and date.

So, anyone with the Micetro Requester access can raise changes, get them approved, and have them go live in Azure DNS, all without having to know anything about Azure and without needing any Azure portal or Azure API access. Micetro's Workflow module is equally applicable to any of the other supported DNS servers and cloud providers.

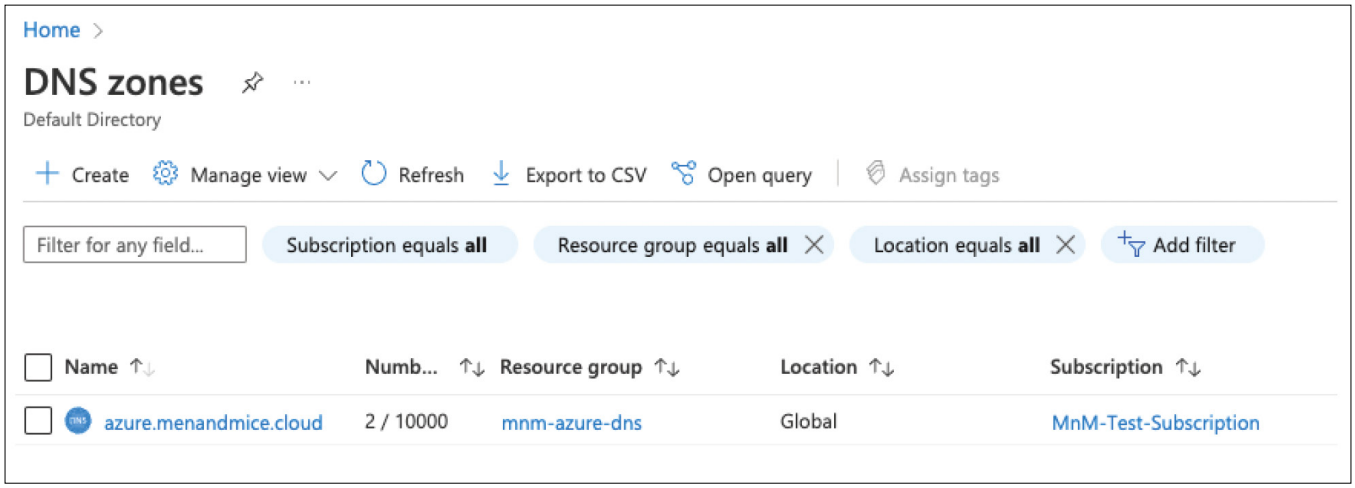
How to manage DNS in Azure?

How do I create a DNS Zone in Azure?

From the Micetro web application, it is trivial to create master zones in Azure DNS. Once you have Micetro DNS administrator privileges, you can go to the "DNS" section and then "Create/master zone". Enter the full Zone name (not forgetting the trailing dot), e.g., "azure.menandmice.cloud.", and ensure you select the correct Master server, which, in our case, will affect the Resource Group "mnm-azure-dns" on "Azure-Multicloud":



To validate this, once submitted, you can see that the zone has been created in Azure DNS (below) including the SOA and NS records.

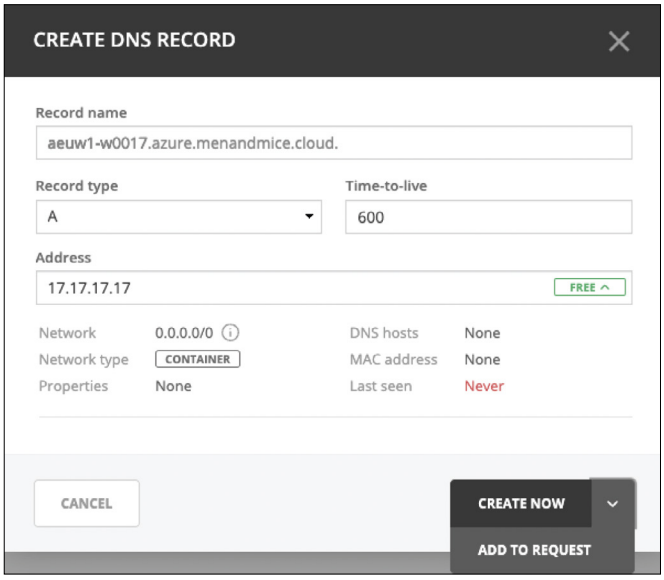


Tip: If you delete and recreate a zone in Azure DNS, always remember to check your NS and SOA records are correct if you are delegating from outside of Azure DNS.

In Micetro, you can subsequently perform actions on the zone such as "Open", "Migrate", and "Delete" while also assigning fine-grained access permissions via Micetro's RBAC by selecting "Access". You can also access the full administrative history of the zone.

How to change DNS in Azure

Now, let's create some records in our newly created zone of "azure.menandmice.cloud.". As a DNS administrator, we can choose to create and action a request immediately, or as a Requester (built-in), we can submit this record creation as a change request.



Once it has been approved and actioned (if you're a DNS Administrator, you can activate it immediately), we can see that the record has been created in Micetro:

NAME	TTL	TYPE	DATA
azure.menandmice.cloud.	2d	NS	ns1-36.azure-dns.com.
azure.menandmice.cloud.	2d	NS	ns2-36.azure-dns.net.
azure.menandmice.cloud.	2d	NS	ns3-36.azure-dns.org.
azure.menandmice.cloud.	2d	NS	ns4-36.azure-dns.info.
aeuw1-w0017.azure.menandmice.cloud.	10m	A	17.17.17.17

And also in Azure DNS:

aeuw1-w0017

Name	Type	TTL	Value	Alias resource type	Alias target
aeuw1-w0017	A	600	17.17.17.17		

So, from a single UI (or API) we can administer records across all our cloud services and DNS servers.

How to create a virtual network (VNet) in Azure

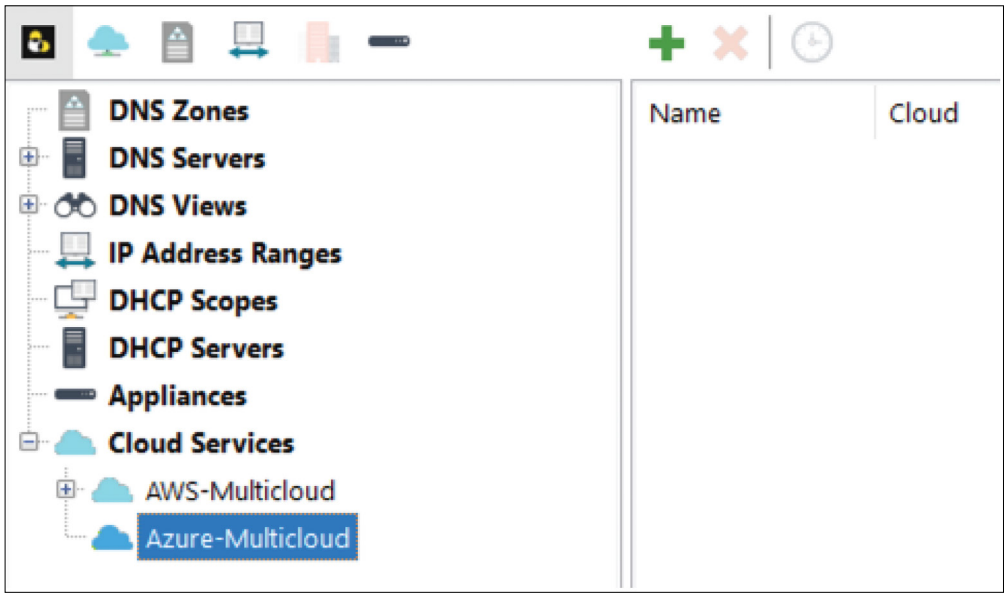
Creating VNets (Virtual Networks)

Micetro makes creating VNets easy and will also automatically track our IP allocations via the IPAM. Let's create

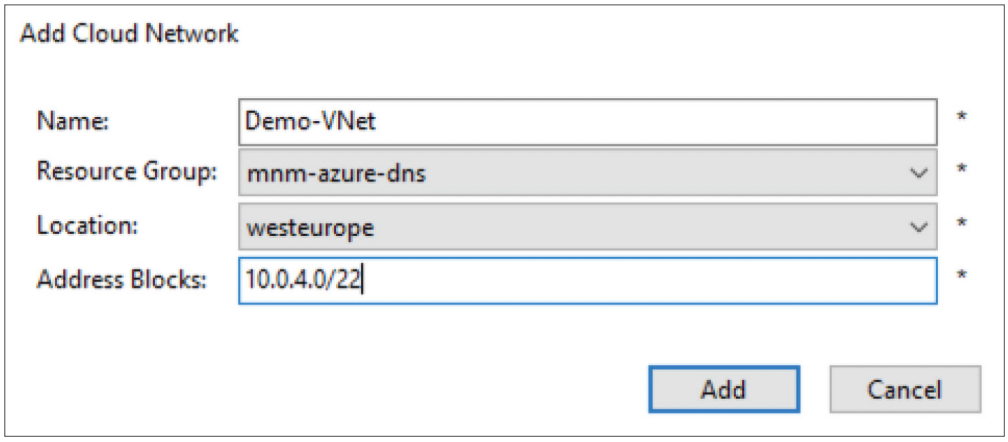
a VNet in the "europewest" Azure region using the CIDR block of "10.0.4.0/22," from which we will allocate two separate subnets of "10.0.4.0/24" and "10.0.5.0/24" from the lower /23.

Note: Currently, we use the Men&Mice Console to create VNets in Azure, but shortly this will become available in the web application, including being applicable for the Workflow module.

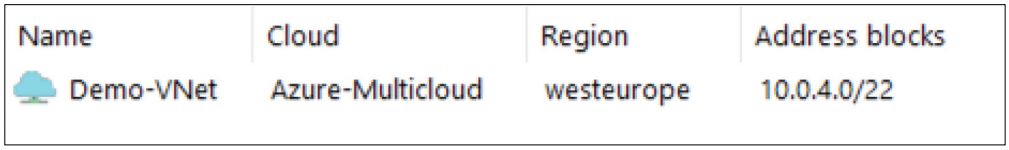
From the Console, select the appropriate cloud service and then use the green plus (or Ctrl+n) to add a new Cloud Network:



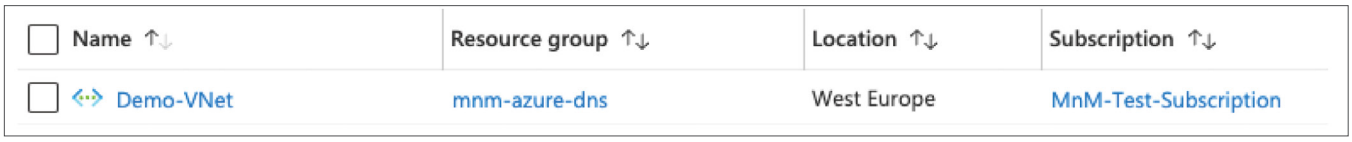
Enter the Name, Resource Group, Location, and Address Blocks required and click Add:



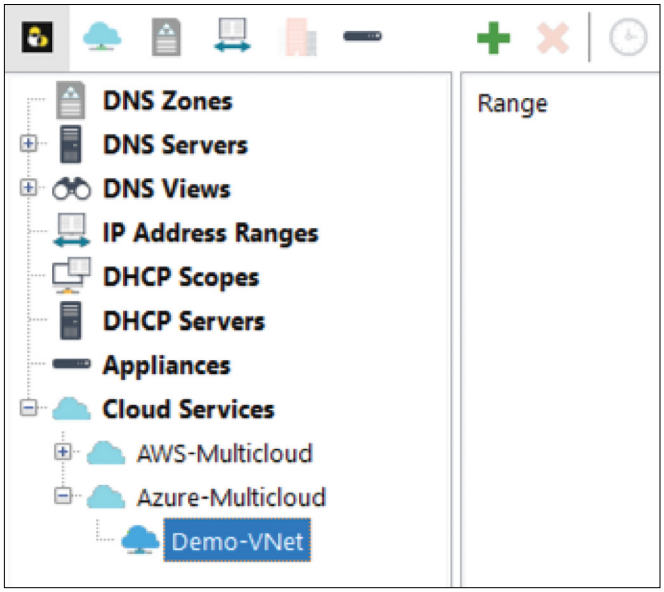
And we can now see our new Demo-VNet listed in Micetro:



But we can also see Demo-VNet is now available and ready for action in Azure:



We will now create two subnets inside our newly formed Demo-VNet. We use the green plus button (or Ctrl+n) again to add the details we want for each of the new subnets using the details below:



Details (subnet 1):

- Subnet: 10.0.4.0/24
- Title: Untrusted-DMZ-Tier
- Description: This is our example initial DMZ tier within the VNet.
- Cloud Network: Demo-VNet
- Address Range: Reserve Network and Broadcast Address

Details (subnet 2):

- Subnet: 10.0.5.0/24
- Title: Trusted-DB-Tier
- Description: This is our example trusted DB tier within the VNet.
- Cloud Network: Demo-VNet
- Address Range: Reserve Network and Broadcast Address

This results in our subnets also going live in Azure:

Name	IPv4	IPv6	Available IPs	Delegated to	Security group	Route table	
Untrusted-DMZ-Tier	10.0.4.0/24	-	251	-	-	-	•••
Trusted-DB-Tier	10.0.5.0/24	-	251	-	-	-	•••

Meanwhile, we can see that our subnets are also being tracked in our unified Micetro IPAM:

10.0.4.0/24	RANGE	2%	Demo-VNet	Untrusted-DMZ-Tier	This is our example initial DMZ tier within the VNet.
10.0.5.0/24	RANGE	2%	Demo-VNet	Trusted-DB-Tier	This is our example trusted DB tier within the VNet.

How can I use the Micetro API to Change DNS in Azure?

Micetro REST API authentication

The Micetro REST API uses basic HTTP user authentication. It then uses the account privileges configured for that user. Privileges are set via the Micetro RBAC in Access Management. It is recommended to create a separate user account for use with the API.

Tip: The API user account should be scoped to only have the rights to perform the tasks required. It should not be re-used or shared to facilitate access from multiple platforms. Create a separate account per use case, role, or system to minimize failure domains.

Example API call(s)

You can test the Micetro REST API using cURL with the appropriate user credentials to administer and access objects.

Tip: When using methods other than the default HTTP GET, remember to supply the header Content-Type of "application/json". We will use cURL to demo some API calls. cURL is available by default in macOS and most Linux distributions, but it is also available as a binary for Windows if you're using Windows and don't want to use Powershell.

Let's use the Micetro API to look up the earlier IPAM record (17.17.17.17) we created (when we added an A record to the "aws.menandmice.cloud." zone):

Using macOS/Linux:

```
curl -silent --user <user>:<password> -X GET \
"https://<your_host>/mmws/api/IPAMrecords/17.17.17.17"
```

Using Windows Powershell:

```
$cred = Get-Credential
Invoke-RestMethod -Method GET -Cred $cred -Uri
"https://<your_host>/mmws/api/IPAMrecords/17.17.17.17" | ConvertTo-Json
-Depth 5
```

Which results in the following JSON blob response containing all the associated data for the IP record (including its DNS records and DHCP information if applicable):


```
{
  "result": {
    "ipamRecord": {
      "addrRef": "IPAMRecords/36",
      "address": "17.17.17.17",
      "claimed": false,
      "dnsHosts": [
        {
          "dnsRecord": {
            "ref": "DNSRecords/64",
            "name": "aeuw1-w0017.azure.menandmice.cloud.
[mnm-azure-dns]",
            "type": "A",
            "ttl": "600",
            "data": "17.17.17.17",
            "comment": "",
            "enabled": true,
            "dnsZoneRef": "DNSZones/12",
            "customProperties": {}
          },
          "ptrStatus": "Unknown",
          "relatedRecords": []
```

```
        }
      ],
      "dhcpReservations": [],
      "dhcpLeases": [],
      "discoveryType": "None",
      "lastSeenDate": "",
      "lastDiscoveryDate": "",
      "lastKnownClientIdentifier": "",
      "device": "",
      "interface": "",
      "ptrStatus": "Unknown",
      "extraneousPTR": false,
      "customProperties": {},
      "state": "Assigned",
      "usage": 4
    }
  }
}
```


Now, let's look for the related A record in DNS by specifying the zone and only using a partial search term:

Using MacOS/Linux:

```
curl -silent --user <user>:<password> -X GET \
"https://<your_host>/mmws/api/DNSZones/azure.menandmice.cloud./DNSRecord
s?filter=type=A AND name=@aeu"
```

Or Windows Powershell:

```
$cred = Get-Credential
Invoke-RestMethod -Method GET -Cred $cred -Uri
"https://<your_host>/mmws/api/DNSZones/azure.menandmice.cloud./DNSRecord
s?filter=type=A AND name=@aeu" | ConvertTo-Json -Depth 5
```

Which finds (1) record returned:


```
{
  "dnsRecords": [
    {
      "name": "aeuw1-w0018",
      "type": "A",
      "ttl": "600",
      "data": "17.17.17.18",
      "enabled": false,
      "aging": 0,
      "dnsZoneRef": "DNSZones/12"
    }
  ],
  "saveComment": "Created by API"
}
```

Now, let's create a totally new DNS A record in our zone "azure.menandmice.cloud." using the name "aeuw1-w0018" and the IP address "17.17.17.18", and then let's check it has propagated to Azure DNS and beyond!

First we shall create a small JSON blob in a file called "new.json":

```
{
  "dnsRecords": [
    {
      "name": "aeuw1-w0018",
      "type": "A",
      "ttl": "600",
      "data": "17.17.17.18",
      "enabled": false,
      "aging": 0,
      "dnsZoneRef": "DNSZones/12"
    }
  ],
  "saveComment": "Created by API"
}
```

Now let's send this data from "new.json" to our endpoint "DNSRecords":

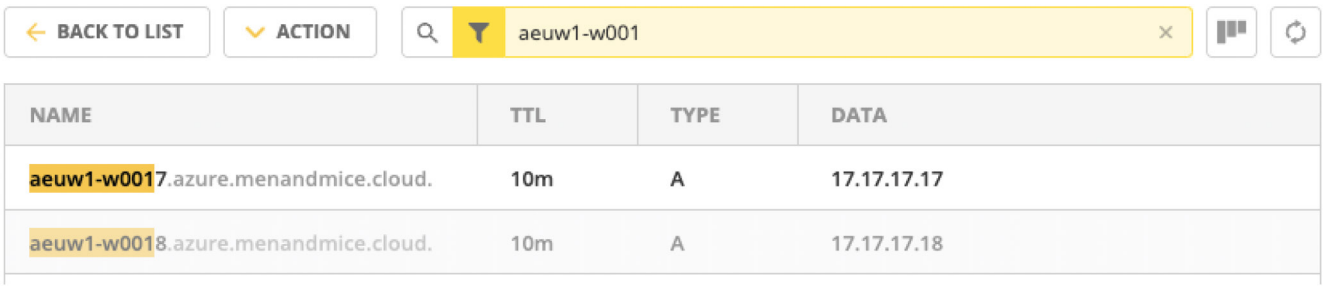
Using MacOS/Linux:

```
curl --header "Content-Type: application/json" \
  --request POST \
  --user <user>:<password> \
  -d @new.json \
  https://<your_host>/mmws/api/DNSRecords
```

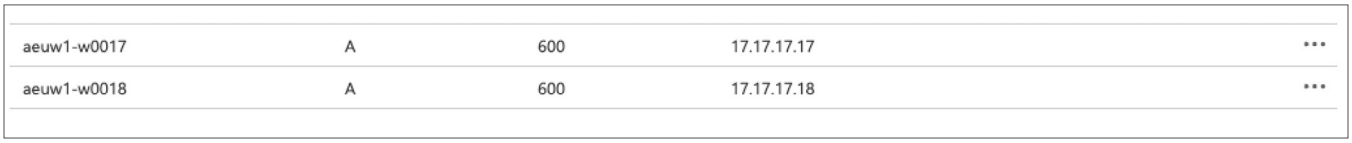
Which should return a HTTP "201 Created" status and the Object Reference (objRef) with no listed errors. We are now pretty happy that a record was indeed created.

```
{"result":{"objRefs":["DNSRecords/65"],"errors":[]}}
```

Now, let's check in Micetro, Azure DNS, and also from the Internet's perspective:
Micetro looks good:



Azure DNS looks good:



And the Internet says yes!

```
dig @8.8.8.8 +short aeuw1-w0018.azure.menandmice.cloud
17.17.17.18
```

Available endpoints and documentation

Once you install the free trial, there is a full set of OpenAPI (Swagger) documentation available directly on your Men&Mice web application accessed via the path /mmws/api/doc/, and the latest product guides are always available at https://menandmice.com/docs.

Corporate Headquarters

4100 Yonge St. 3rd Floor, Toronto, ON, M2P 2B5
1-416-646-8400 | 1-866-895-6931

bluecat.com

Next steps

Use an overlay and orchestration solution to regain network visibility and control.

Request a free trial