Managing VPC Creation and IP space for Organizations at scale

Large organizations have challenges creating VPCs within their LZ/CT while managing IP address space. Networking teams traditionally have no automation to provision new VPCs in each child account while also tracking CIDR blocks used. This blog post outlines how organizations and service catalog can be leveraged in partnership with Infoblox DDI to simplify VPC provisioning with no manual interaction for networking teams. The solution delivers enhanced visibility in to the governance of infrastructure at scale while reducing workload and provisioning time for newly vended accounts.

This blog post will walk through

* Set up of the Infoblox Grid including containers and extensible attributes.
* the architecture and code in the central account responsible for communicating with Infoblox for new IP Address ranges for VPCs.
* the architecture and code in the customer account as part of a service catalog portfolio responsible for the creation of the VPC.

# Solution overview

The solution uses the following services:

* [Infoblox DDI for AWS](https://www.infoblox.com/products/ddi-for-aws/)
* [AWS Organizations](https://aws.amazon.com/organizations/)
* [AWS Lambda](https://aws.amazon.com/organizations/)
* [AWS Key Management Service](https://aws.amazon.com/kms/)
* [AWS Secrets Manager](https://aws.amazon.com/secrets-manager/)
* [AWS Simple Notification Service](https://aws.amazon.com/sns/)
* [AWS Virtual Private Cloud](https://aws.amazon.com/vpc/)
* [AWS Transit Gateway](https://aws.amazon.com/transit-gateway/)
* [AWS CloudFormation](https://aws.amazon.com/cloudformation/)
* [AWS Service Catalog](https://aws.amazon.com/servicecatalog/)
* [AWS Identity and Access Management](https://aws.amazon.com/iam/)
* [AWS Resource Access Manager](AWS%20Identity%20and%20Access%20Management)

A designated networking account contains a centrally managed transit gateway which is shared to all eligible child accounts using AWS RAM.

A designated central account contains a VPC with access to the Infoblox Grid Master. A lambda function is deployed to this central account within the VPC to process requests from child accounts that are received through a SNS topic. This SNS topic has a topic policy that allows any account from the organization to publish messages to this topic. The lambda function uses AWS Secrets Manager to store the authorization information used to authenticate calls with the Infoblox Grid. The SNS topic and Secrets Manager secret are protected by a customer managed AWS KMS Key.

Child accounts throughout the organization access a product that is shared as part of a portfolio using AWS Service Catalog. This CloudFormation template creates a

* VPC
* 2 private subnets
* A transit gateway attachment spanning 2 private subnets
* Private subnet route table containing a default route to the TGW
* A lambda function based custom resource which checks that
	+ only a single TGW exists in the account
* A custom resource that has a service token referencing the SNS topic in the central account.

If multiple Infoblox network containers are allocated for a single AWS region, the child account will receive the CIDR block information of the next available network from a randomly selected free network container.

These elements work in tandem to deliver a seamless experience as new accounts are created throughout your organization. Individual account owners can create a VPC that deliver connectivity through your organization’s TGW without overlapping CIDR Blocks. This not only reduces the provisioning time for accounts requiring access to VPCs, it increases governance and visibility as Infoblox DDI can identify which CIDR blocks have been assigned for an account in a region.



[ALT TEXT] VPC Service Catalog Product Architecture Described Above

# Setting up AWS Transit Gateway share

1. Follow the steps listed [here](https://docs.aws.amazon.com/vpc/latest/tgw/tgw-transit-gateways.html#create-tgw) to create a transit gateway according to your organization’s needs.
2. Follow the steps listed [here](https://docs.aws.amazon.com/vpc/latest/tgw/tgw-transit-gateways.html#tgw-sharing) to share the transit gateway across your organization by specifying an AWS account, OU, or organization

# Setting up Infoblox grid master

Connect to your organization’s existing Infoblox DDI Grid Master (hosted on-prem or in the cloud) to be accessible through the VPC located in the central account. If you do not have access to an existing Infoblox appliance, follow the steps listed [here](https://www.infoblox.com/wp-content/uploads/infoblox-deployment-guide-deploy-infoblox-vnios-instances-for-aws.pdf) to deploy a new Infoblox appliance on AWS.

## Setting up Required Infoblox Extensible Attributes

This solution uses 'Cloud Account ID', 'Region', and 'Cloudformation Stack ID' as extensible attributes for IPv4 networks. 'Region' is used as an extensible attribute for IPv4 Network Containers. Be sure that the 'Restricted to Objects' field for these extensible attributes are unrestricted or allow these object types.

These can be automatically configured by keeping the default parameter of 'Yes' on the 'Setup Infoblox Extensible Attribute' parameter. Alternatively, follow the Infoblox guide [here](https://docs.infoblox.com/display/nios84/Managing%2BExtensible%2BAttributes#ManagingExtensibleAttributes) to add or edit these manually.

## Setting up Infoblox Network Container

The parent CIDR Block(s) that this solution will create child VPC CIDR Blocks from is contained in the Infoblox grid as IPv4 Network Containers that have an extensible attribute of 'Region' with the AWS region (i.e. us-east-1) as the value.

These can be automatically configured by supplying a comma delimited list of CIDR Blocks in CIDR notation to the 'Infoblox Network Containers' parameter. Alternatively, follow the Infoblox guide [here](https://docs.infoblox.com/display/nios85/Configuring%2BIPv4%2BNetworks#ConfiguringIPv4Networks-bookmark2343) to add or modify these manually.

# Central account setup

Be sure to have

* provisioned a VPC in this account that contains access to a minimum of 2 private subnets in different availability zones that allow connectivity to AWS service endpoints. This can be achieved commonly using NAT Gateways or VPC endpoints.
* Network interfaces in the subnets should also have a valid network path to the Infoblox NIOS appliance.

Upload the zip file from Github (ADD HYPERLINK LATER) to a S3 bucket in your account. Follow this [guide](https://docs.aws.amazon.com/AmazonS3/latest/user-guide/upload-objects.html) to learn more about file uploads to S3.

Download the CloudFormation template from GitHub (ADD HYPERLINK LATER) and create a CloudFormation stack. For more information about how to create a CloudFormation stack, see [Getting Started with AWS CloudFormation](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/GettingStarted.html) in the AWS CloudFormation User Guide.

Most of the parameters can be left as default, but some may need to be modified depending on organizational preference. The lambda security group may use the default VPC security group if the outbound rules allow traffic to the Infoblox appliance. Changing parameters such as the SNS Topic Name may require changes to the corresponding template deployed in the service catalog in child accounts.

After your stack has finished deploying, navigate to the **Resources tab** and select the hyperlink for the Physical ID associated **rSecretsManagerSecret** Logical ID.



[ALT TEXT] screenshot of Cloudformation console showing the resources tab of the deployed cloudformation tempalate

This will bring you to the AWS Secrets Manager Console. Click **Retreive secret value**



[ALT TEXT] screenshot of AWS Secrets Manager console showing the newly created secret

Click **Edit** in the Secret value section.

****

[ALT TEXT] screenshot of the Secret value section of the AWS Secrets Manager console

In the **Edit secret value** header, choose **Plaintext** and replace all the text in the text field with your secure Infoblox Password. This password in combination with the username entered as a AWS CloudFormation parameter above should be for an account that has been provisioned for use in Infoblox DDI. Follow the Infoblox [documentation](https://docs.infoblox.com/display/nios85/Managing%2BAdministrators) for more information about required permissions to manage the grid and local authentication/Active Directory users. Click the orange **Save** button.

 ****

[ALT TEXT] screenshot of the Edit secret value section of the AWS Secrets Manager console

# Setting up the AWS Service Catalog product in child accounts

1. Create a service catalog portfolio in a designated administrator account following the instructions [here](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/portfoliomgmt-create.html)
2. Download the CloudFormation template from Github (ADD LINK HERE Later) and follow this [guide](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/portfoliomgmt-products.html) to upload the template as a product to the portfolio
3. Use this [guide](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/portfoliomgmt-constraints.html) to add template constraints that restrict parameters of the product template. This can be used to lock the **Central Account ID** parameter to one designated for your organization.
4. Follow this [guide](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/catalogs_portfolios_users.html) to grant access to users in this account for testing
5. Follow this [guide](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/catalogs_portfolios_sharing_how-to-share.html) to share this portfolio to designated AWS accounts within your organization.

# Cleanup

If you no longer require access to this solution

1. Delete the CloudFormation template deployed in the central account setup section
2. Delete the S3 Bucket and files uploaded in the central account setup section
3. Delete the service catalog product and/or portfolio deployed in the service catalog setup section.
4. Delete the transit gateway in the networking account
5. Decommission the Infoblox appliance by terminating the EC2 instance if hosted on AWS or removing access to the on-prem grid instance.

# Conclusion

You’re now ready to automate new VPC creation throughout your organization reducing new account provisioning time, reducing workload for network administrators while enhancing their visibility in to IP Address Space usage across the organization in partnership with Infoblox DDI.