Networking & Content Delivery

Streamline access to most used AWS services using VPC Endpoints

by Kawsar Kamal and Yuri Duchovny | on 14 MAR 2024 | in AWS PrivateLink, Networking & Content Delivery | Permalink | Share

Amazon Virtual Private Cloud (Amazon VPC) endpoints, powered by Amazon Web Services (AWS) PrivateLink, can be used to privately connect your applications to AWS services as if they were in your VPC. For enterprises that use many AWS services, it may be difficult to understand which services are being used most often and therefore can benefit from using VPC endpoints. In addition, administrators may lack the required visibility to know which AWS services are being called from a VPC.

We propose a solution for identifying the top AWS services being used from your VPCs. By configuring VPC endpoints, traffic to these services will traverse a secure and optimized route that avoids internet gateways and NAT gateways. When working with many VPCs, customers will also benefit from simplified network management. The tools we use in the solution include Amazon Route 53 Resolver query logs, AWS Lambda, Amazon CloudWatch Logs, and optionally Amazon VPC Flow Logs.

Prerequisites

To deploy this solution, you need access to an AWS account with permissions to create and administer the following resource types (Read Changing permissions for an IAM user on how to set up the appropriate AWS Identity and Access Management (IAM) permissions for these services.):

- Amazon VPC
- Amazon CloudWatch Logs
- IAM role and policies
- Lambda functions
- Amazon Route 53 Resolver
- Amazon VPC flow logs

To follow along with the steps you will also need access to the AWS Management Console.

Solution overview

By default, DNS requests that originate within your Amazon VPC use the Amazon Route 53 Resolver to resolve queries. You can log all your VPC DNS queries with a feature of Route 53 called Route 53 Resolver query logs. Amazon VPC flow logs enable you to capture information about IP traffic going to and from network interfaces inside a VPC. Both Amazon VPC flow logs and Route 53 Resolver query logs allow you to publish log data in a variety of destinations, including:

- CloudWatch Logs

- Amazon Simple Storage Service (Amazon S3) buckets
- Amazon Kinesis Data Firehose

In this solution, we configure a log group in CloudWatch Logs as the destination for both Amazon VPC flow logs and Route 53 Resolver query logs. We query the log group using CloudWatch Logs Insights, which is a feature that allows you to search and analyze log data in CloudWatch Logs with a purpose-built query language.

The solution uses the following workflow to identify the most frequently used AWS services in an Amazon VPC that may benefit from VPC endpoints:

1. Identify the most used AWS services using VPC DNS Queries – Enable Route 53 Resolver query logs and use a CloudWatch Logs Insights query to check for the most frequent DNS queries against the **amazonaws.com** namespace.
2. Estimate traffic volume to AWS services (Optional) – Configure flow logs and analyze using CloudWatch Logs Insights to gather the number of bytes being sent to IP addresses identified from Step 1. Flow logs can be verbose and incur additional log storage cost. We therefore suggest it as an optional step.
3. Review other considerations – Review considerations such as Amazon VPC endpoint costs and security. This is a manual step for any qualitative decisions before proceeding with endpoint creation.
4. Implement Amazon VPC endpoints – Create VPC endpoints for AWS services identified in the previous steps.

Solution walkthrough

The following diagram shows the solution workflow and components. We have included a set of AWS CloudFormation templates to help automate the overall workflow.

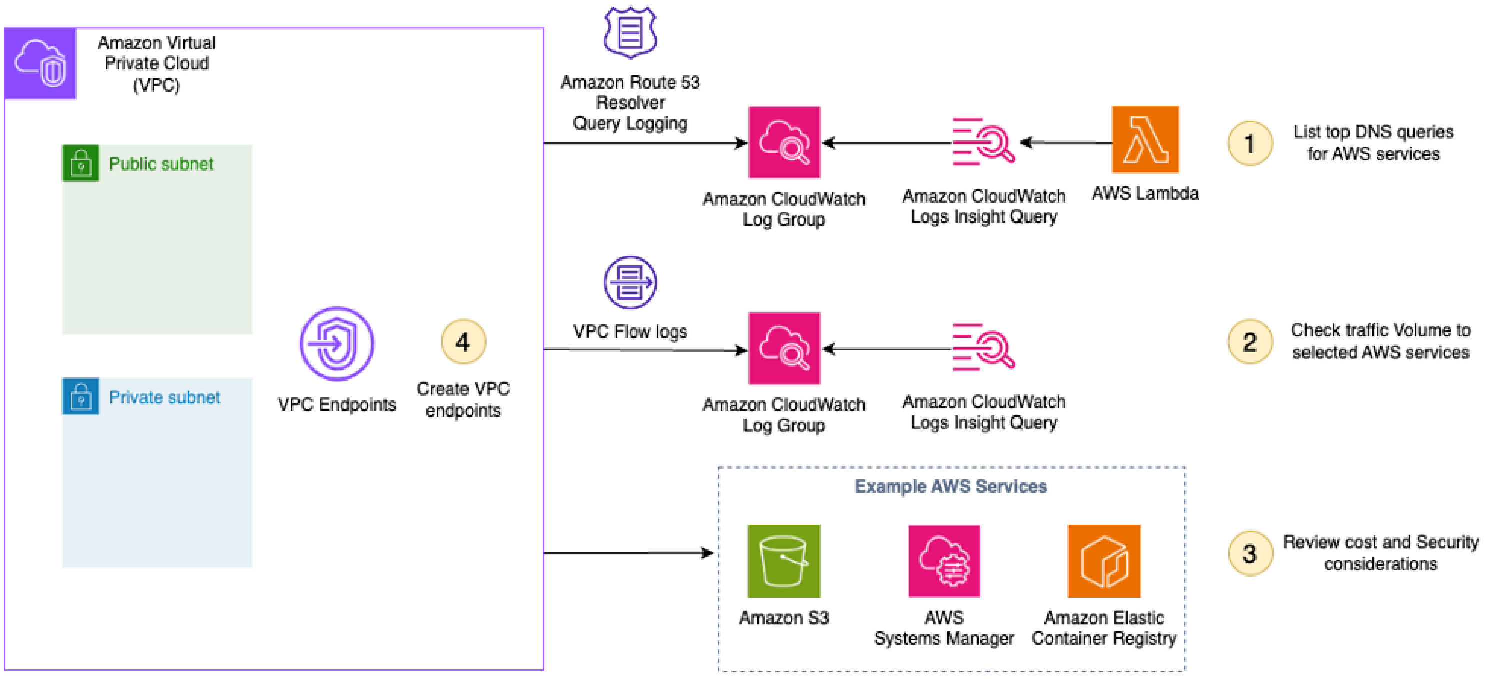

Amazon Virtual Private Cloud (VPC)
Public subnet
Private subnet
VPC Endpoints
4
Create VPC endpoints
Amazon Route 53 Resolver Query Logging
Amazon CloudWatch Log Group
Amazon CloudWatch Logs Insight Query
AWS Lambda
1
List top DNS queries for AWS services
VPC Flow logs
Amazon CloudWatch Log Group
Amazon CloudWatch Logs Insight Query
2
Check traffic Volume to selected AWS services
Example AWS Services
Amazon S3
AWS Systems Manager
Amazon Elastic Container Registry
3
Review cost and Security considerations

Figure 1: Solution workflow and components

Step 1. Identify the most used services using Amazon VPC DNS queries

In this step, we enable Route 53 Resolver query logs and configure a CloudWatch log group as its destination using a CloudFormation template. The template also creates a Lambda function that queries Route 53 Resolver query logs to help identify AWS services with the highest number of VPC DNS queries.

There are two CloudFormation templates based on whether you are deploying the solution in a new or existing VPC:

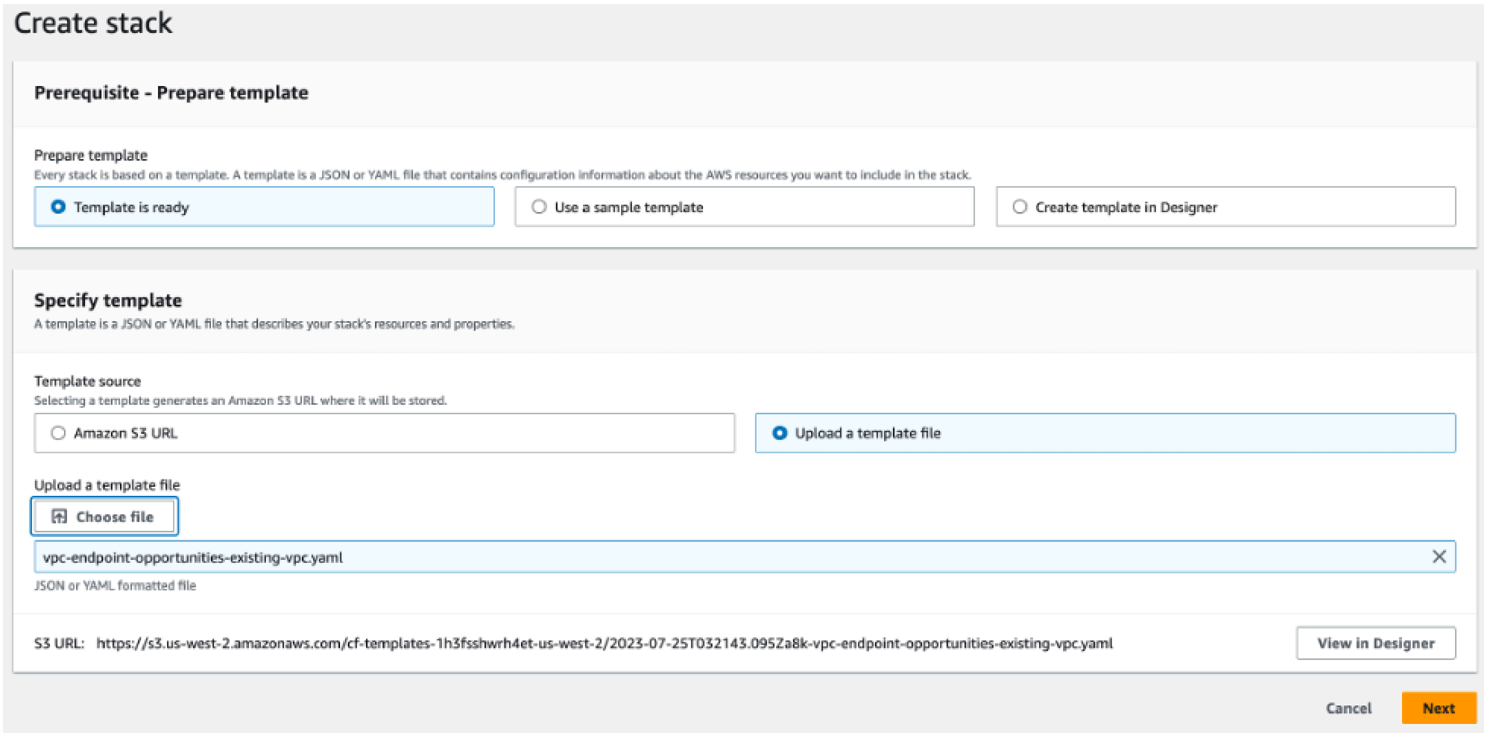
1. aws-vpc-endpoints-exising-vpc.yaml – This template deploys the solution inside a previously created Amazon VPC. Use this template to evaluate an existing VPC with deployed applications.
2. aws-vpc-endpoints-new-vpc.yaml – This template creates a new Amazon VPC and deploys the solution in that VPC. After launching the stack, you will need to create some traffic from the newly created VPC to test the solution. For example, you may launch an Amazon Elastic Compute Cloud (Amazon EC2) instance and access a S3 bucket from the instance.

Download one of the preceding templates into a working directory, then proceed with section 1.1 or 1.2 based on whether you want to deploy the solution to a new or existing VPC, respectively. We use the AWS Management Console to deploy the template. Refer to the CloudFormation Getting Started guide for step-by-step guidance on deploying a template using the AWS Management Console. If you prefer to use the AWS Command Line Interface (AWS CLI) instead, reference this documentation: Using the AWS Command Line Interface.

1.1 Deploy to an existing VPC

From the CloudFormation dashboard, choose the aws-vpc-endpoints-existing-vpc.yaml template file to begin the stack creation workflow.

Select **Next** to proceed to the **Specify stack details** page. Provide a stack name, such as vpc-endpoints-blog, and a VPC ID for the VPCID parameter.


Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.
[✓ Template is ready] Use a sample template Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

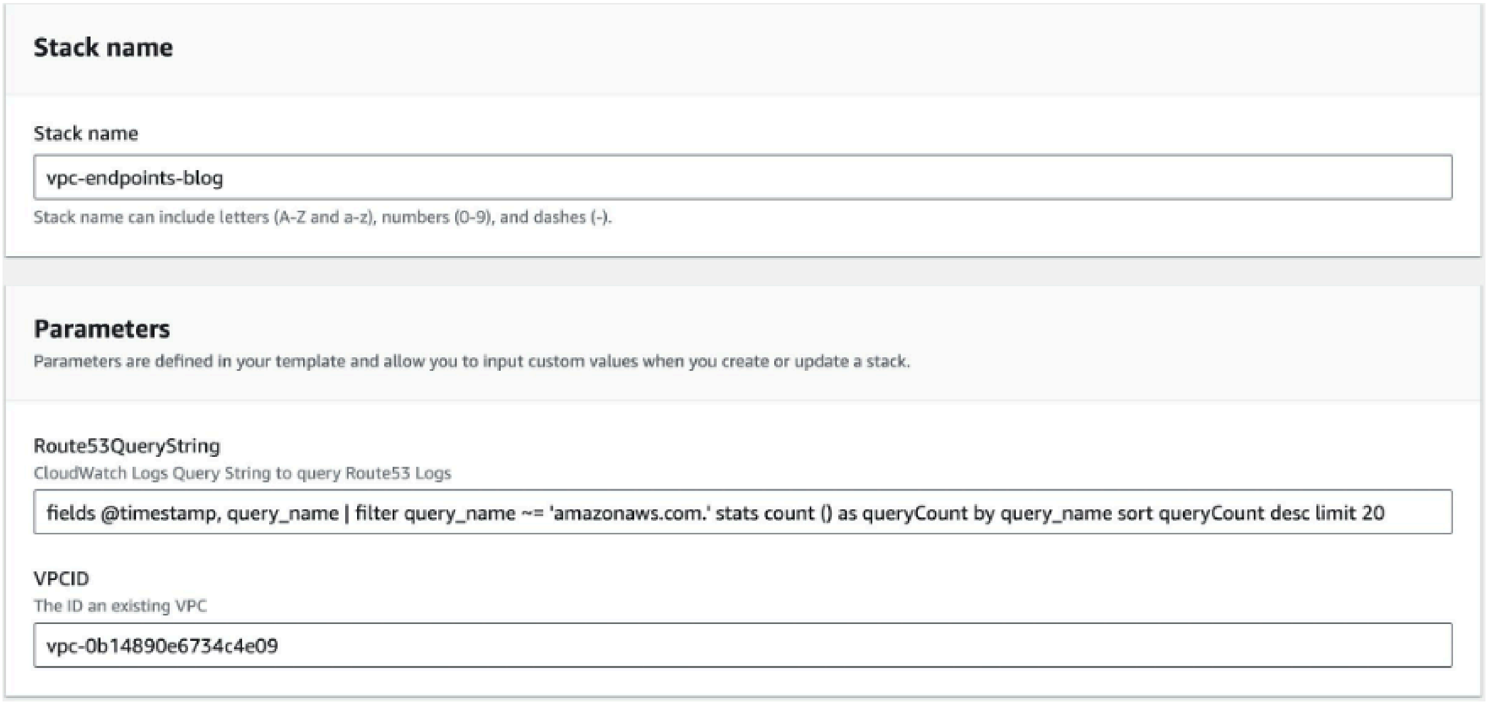
Template source
Selecting a template generates an Amazon S3 URL where it will be stored.
Amazon S3 URL [✓ Upload a template file]

Upload a template file
Choose file
vpc-endpoint-opportunities-existing-vpc.yaml
JSON or YAML formatted file

S3 URL: https://s3.us-west-2.amazonaws.com/cf-templates-1h3fsshwrh4et-us-west-2/2023-07-25T032143.095Za8k-vpc-endpoint-opportunities-existing-vpc.yaml

Figure 2: Create CloudFormation Stack from existing template

Select **Next** and continue with the remaining steps to deploy this stack. Proceed to step 1.3 once you have deployed the stack.


Stack name

Stack name
vpc-endpoints-blog
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

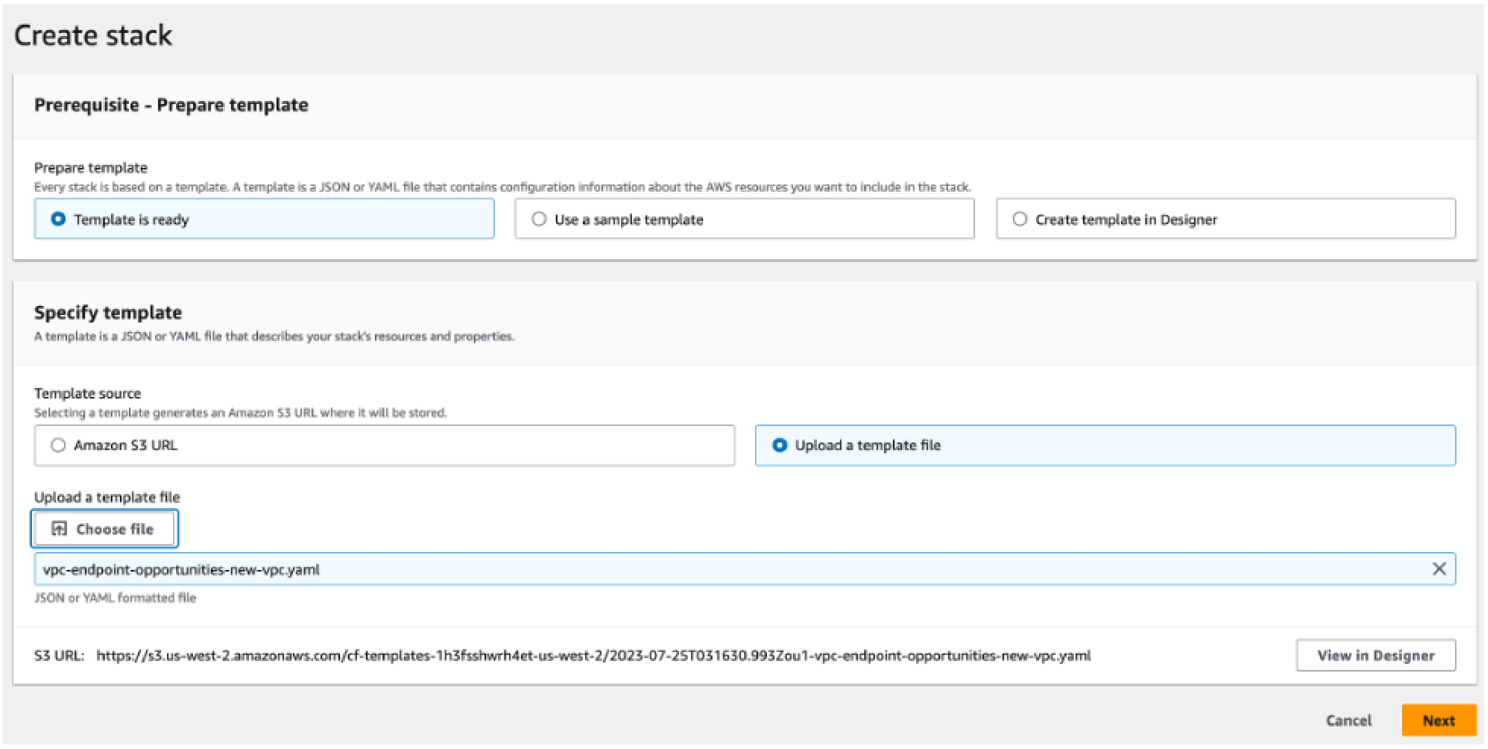
Route53QueryString
CloudWatch Logs Query String to query Route53 Logs
fields @timestamp, query_name | filter query_name ~= 'amazonaws.com.' stats count () as queryCount by query_name sort queryCount desc limit 20

VPCID
The ID an existing VPC
vpc-0b14890e6734c4e09

Figure 3: Specify Stack details

1.2 Deploy to a new VPC

From the CloudFormation dashboard, choose the aws-vpc-endpoints-new-vpc.yaml template file to begin the stack creation workflow.


Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

[✓ Template is ready] Use a sample template Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

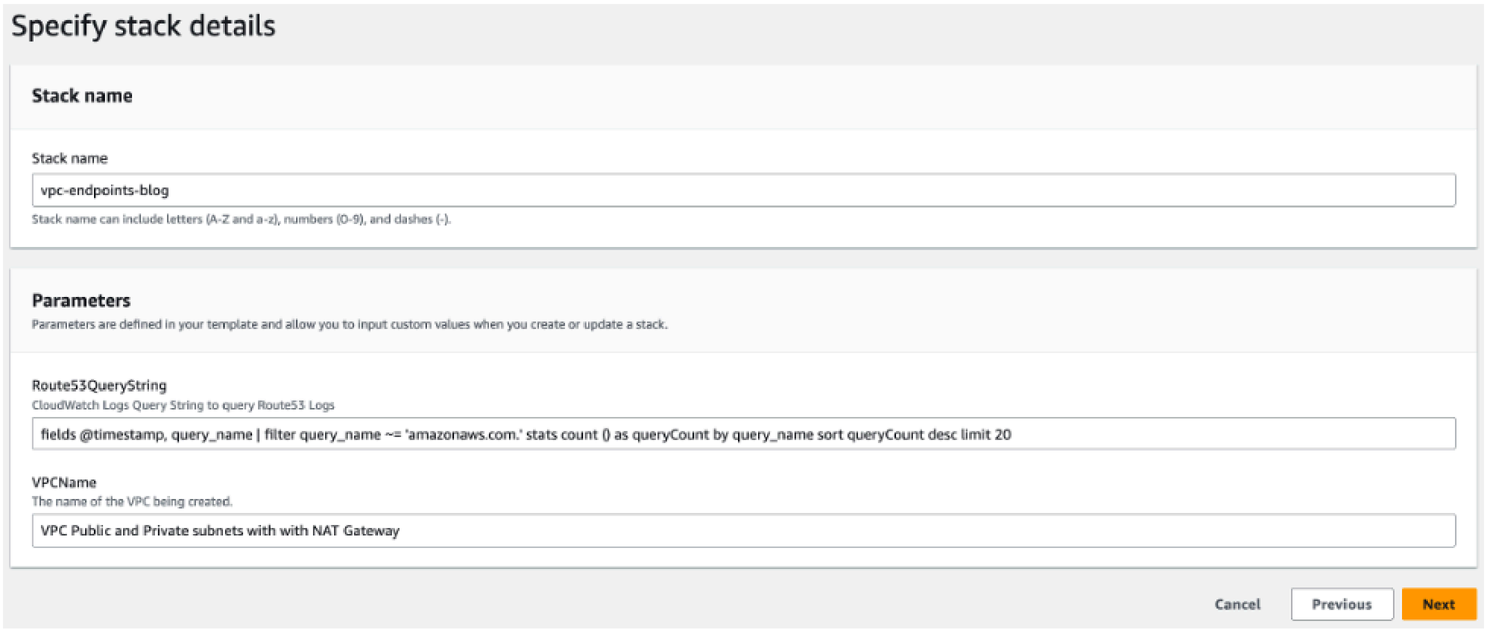
Amazon S3 URL [✓ Upload a template file]

Upload a template file
Choose file
vpc-endpoint-opportunities-new-vpc.yaml
JSON or YAML formatted file

S3 URL: https://s3.us-west-2.amazonaws.com/cf-templates-1h3fsshwrh4et-us-west-2/2023-07-25T031630.993Zou1-vpc-endpoint-opportunities-new-vpc.yaml

Figure 4: Create CloudFormation Stack from existing template

Choose the **Next** button to proceed to the Specify stack details page. Provide a stack name, such as vpc-endpoints-blog, and keep default values for the Parameters. Select **Next** and continue with the remaining steps using the default options to deploy this stack.


Specify stack details

Stack name

Stack name
vpc-endpoints-blog
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Route53QueryString
CloudWatch Logs Query String to query Route53 Logs
fields @timestamp, query_name | filter query_name ~= 'amazonaws.com.' stats count () as queryCount by query_name sort queryCount desc limit 20

VPCName
The name of the VPC being created.
VPC Public and Private subnets with with NAT Gateway

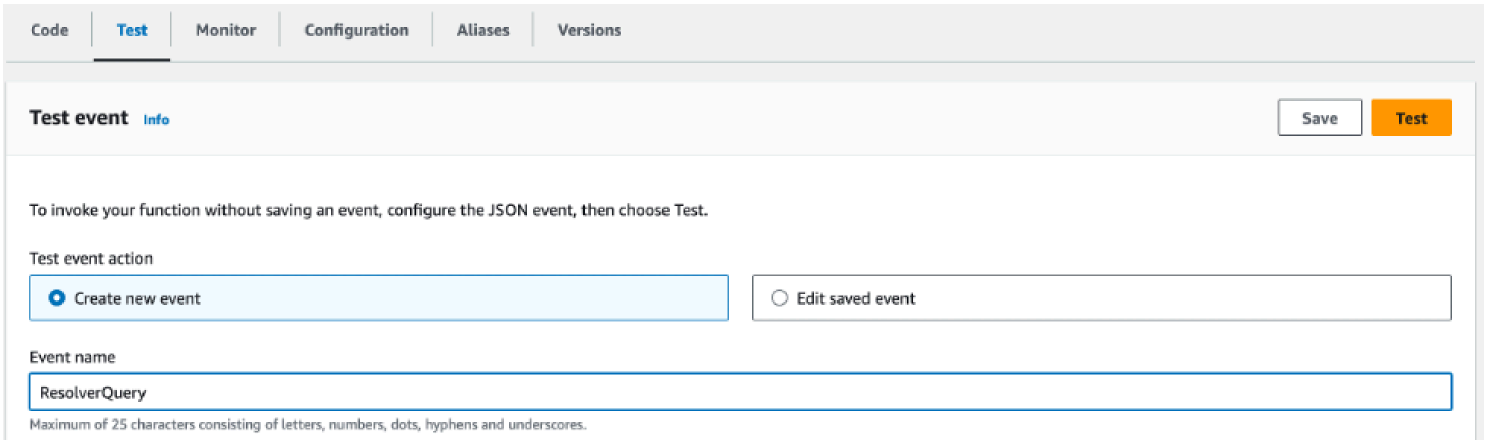
Figure 5: CloudFormation Specify stack details form

1.3 Query DNS logs using CloudWatch Logs Insights query

In this section, we use CloudWatch Logs Insights query to find which AWS services are being queried most often. The CloudFormation stack in the previous step creates a log group in CloudWatch Logs with a seven day retention period as the destination for Route 53 Resolver query logs. Before querying the logs, you may need to wait for a minimum time period to gather sufficient DNS queries from applications and resources in your VPC. The appropriate time period will depend on how often AWS APIs are being invoked from your VPC.

The CloudFormation stack in the preceding section also deploys a Lambda function to query Route 53 Resolver query logs. Navigate to the Lambda dashboard in the console and choose the function that has **CWLogsQueryLambda** as part of its name. Use the following steps to execute the function:

- Select the **Test** tab and enter “ResolverQuery” in the Event name textbox
- Choose the **Save** button
- Choose the **Test** button



To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

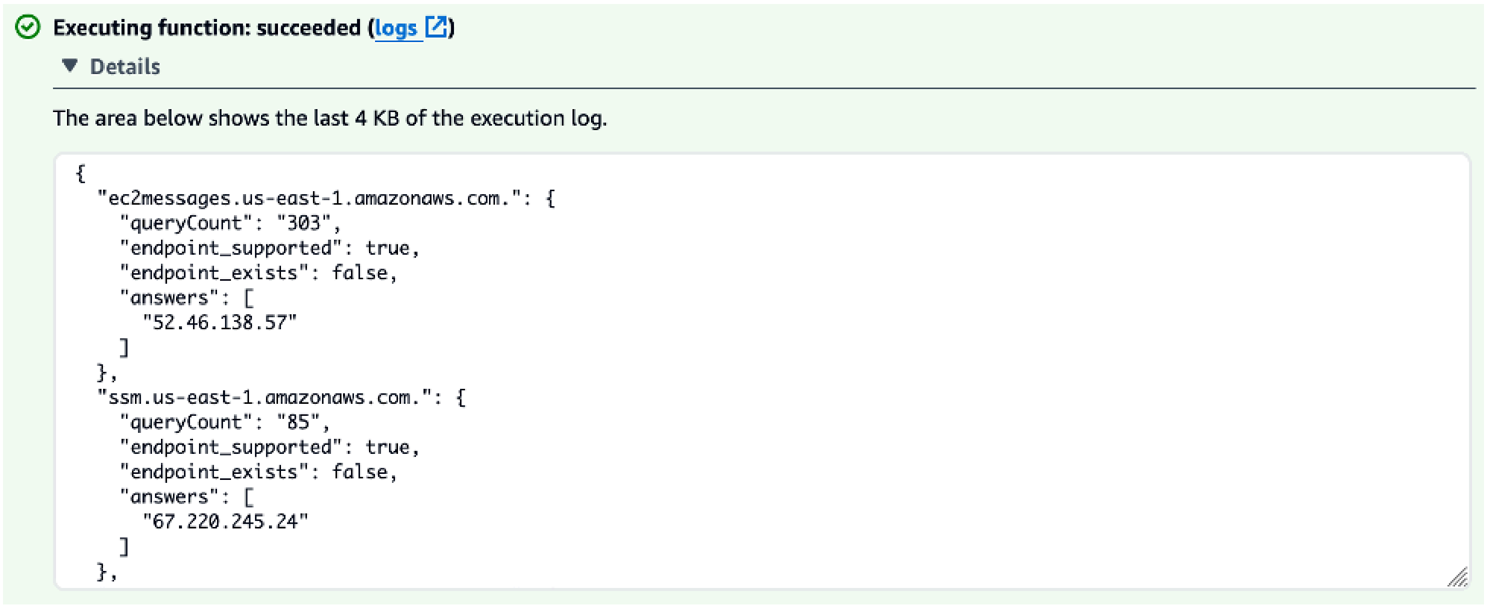
(● Create new event) ○ Edit saved event

Event name: ResolverQuery

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Figure 6: Use a test event to invoke Lambda function

Allow the function execution to complete and choose the Logs link to review the full output. For subsequent invocations, you can use the previously saved event called ResolverQuery to invoke the Lambda function.



The area below shows the last 4 KB of the execution log.

```
{
  "ec2messages.us-east-1.amazonaws.com.": {
    "queryCount": "303",
    "endpoint_supported": true,
    "endpoint_exists": false,
    "answers": [
      "52.46.138.57"
    ]
  },
  "ssm.us-east-1.amazonaws.com.": {
    "queryCount": "85",
    "endpoint_supported": true,
    "endpoint_exists": false,
    "answers": [
      "67.220.245.24"
    ]
  },
```

Figure 7: Results from the CWLogsQueryLambda function

The Lambda function provides a JSON response containing a map of objects. For each object, the key is the AWS service endpoint being accessed. The value is an object with the following attributes:

- `queryCount` : The number of DNS queries to this endpoint
- `endpoint_supported` : Whether a VPC endpoint can be created for this AWS service—the DescribeVpcEndpointServices API is used to obtain this value
- `endpoint_exists` : Whether a VPC endpoint has been created for this AWS service—the DescribeVpcEndpoints API is used to obtain this value.
- `answer` : A list containing the IPv4 responses returned by the DNS query—this will be useful for querying flow logs, as described in the next step.

The objects are sorted by top query counts, which is noted in the field **queryCount**. To review or edit the CloudWatch Logs Insights query being used by the Lambda function review the CloudFormation parameter **Route53QueryString** in the CloudFormation template file.

Step 2: Estimate traffic volume to AWS services

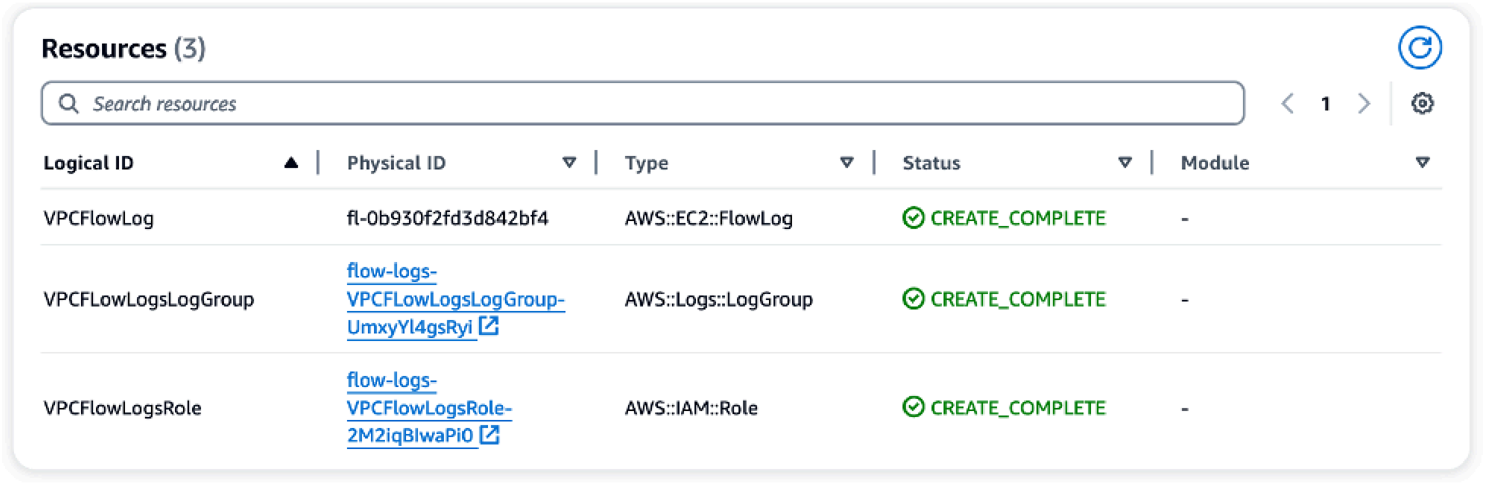
As an optional step, you can use VPC flow logs to estimate the volume of traffic being sent to a specific AWS service. You can also use VPC flow logs to verify whether traffic is being sent to AWS services by using an Amazon VPC endpoint compared to over the internet. After enabling VPC flow logs, we can query it using one or more destination IP addresses.

For more details on configuring flow logs read Create a flow log documentation. Note that flow logs only capture matching traffic from new events, so you will need to wait for some time before logs are made available for querying.

Step 2.1. Enable Amazon VPC flow logs using CloudFormation template

Download this CloudFormation template vpc-flow-logs.yaml in a working directory and use the CloudFormation stack creation workflow similar to step 1 to deploy this template. On the Specify stack details page, provide a stack name such as "vpc-flow-logs" and the relevant VPC ID for the VPCID parameter value.

This stack enables flow logs at a VPC level and configures a log group as the destination with a retention period of seven days. After the stack is deployed, select the Resources tab to view the destination log group. The following screenshot (Figure 8) shows how this looks on the console.


Resources (3)

Search resources

Logical ID	Physical ID	Type	Status	Module
VPCFlowLog	fl-0b930f2fd3d842bf4	AWS::EC2::FlowLog	CREATE_COMPLETE	-
VPCFLowLogsLogGroup	flow-logs-VPCFLowLogsLogGroup-UmxyYl4gsRyi	AWS::Logs::LogGroup	CREATE_COMPLETE	-
VPCFlowLogsRole	flow-logs-VPCFlowLogsRole-2M2iqBlwaPi0	AWS::IAM::Role	CREATE_COMPLETE	-

Figure 8: CloudFormation Stack Resources showing VPC Flow logs log group destination

Step 2.2. Query Amazon VPC flow logs

To estimate the traffic volume being sent to an AWS service, we specify the IP addresses being used to reach that service as part of a CloudWatch Logs Insights query. These were gathered in Step 1.3.

As an example, consider the top two DNS queries we received in Step 1.3:

```
"ec2messages.us-east-1.amazonaws.com.": {
   "queryCount": "303",
   "endpoint_supported": true,
   "endpoint_exists": false,
   "answers": [ "52.46.138.57" ]
},
 "ssm.us-east-1.amazonaws.com.": {
   "queryCount": "85",
   "endpoint_supported": true,
   "endpoint_exists": false,
   "answers": [ "67.220.245.24" ]
}
```

Using the IP addresses in the answers section, we can construct the following CloudWatch Logs Insights query:

```
fields @timestamp, @message, @logStream, @log
  | filter action="ACCEPT" and (dstAddr = "52.46.138.57" or dstAddr = "67.220.245.24")
  | stats sum(bytes) as bytesTransferred by dstAddr
  | sort bytesTransferred desc
  | limit 20
```

Running the preceding query in the VPC flow logs log group will provide the number of bytes sent to the associated AWS service. This is another data point for consideration when creating a VPC endpoint. For example, VPC flow logs may reveal minimal traffic volume to an AWS service even if Route 53 Resolver query logs show a high number of DNS resolutions. The opposite scenario may also be true. A step-by-step guide for running a CloudWatch Logs Insights query is described at: Tutorial: Run and modify a sample query.

Step 3: Review other considerations

Review the following considerations, then continue to the next section to deploy one or more VPC endpoints.

VPC endpoint type

You can access supported AWS services privately from your VPC using an interface VPC endpoint powered by AWS PrivateLink. They are supported by a growing list of AWS services listed here: AWS services that integrate with AWS PrivateLink. Gateway endpoints are another type of VPC endpoint that do not use AWS PrivateLink.

To access Amazon S3, you can use either an interface VPC endpoint or a gateway VPC endpoint. At the time of writing, only gateway VPC endpoints are supported with Amazon DynamoDB. Another key difference is that gateway endpoints do not allow access from on-premises networks, from peered VPCs in other AWS Regions, or through a transit gateway. For those scenarios, you must use an interface endpoint.

Cost

While gateway VPC endpoints have no cost, interface Amazon VPC endpoints do carry a cost. Review the information on AWS PrivateLink pricing and Amazon VPC pricing to learn more about how these services are priced. The AWS pricing calculator can help compare the cost of using Amazon VPC endpoints with the cost of using the NAT gateway and the internet gateway to reach an AWS service.

Security

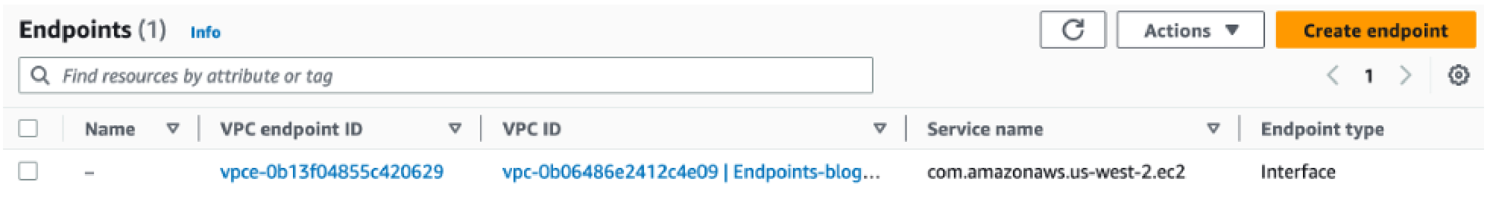
Multiple AWS services integrate with AWS PrivateLink , which supports endpoint policies on the VPC endpoints. An endpoint policy is a resource-based policy that you attach to a VPC endpoint to control which AWS principals can use the endpoint to access an AWS service. You can use the AWS PrivateLink endpoint policies in conjunction with IAM identity-based policies to further tighten access to the AWS services and resources by using VPC PrivateLink endpoints as part of your defense-in-depth approach. To learn more, refer to AWS services that integrate with AWS PrivateLink.

Transitioning to PrivateLink

Even after implementing Amazon VPC endpoints, you may need to continue having a NAT gateway to avoid interruptions to outbound internet access or to AWS services access for which Amazon VPC endpoints are not yet supported or implemented.

Step 4. Creating VPC endpoints

By this step, you should have identified one or more AWS services for which we want to create Amazon VPC endpoints. To check which endpoints have already been created, navigate to Amazon VPC on the console and select Endpoints from the left side navigation pane. As shown in Figure 8, an interface VPC endpoint exists for the Amazon EC2 service.



Name	VPC endpoint ID	VPC ID	Service name	Endpoint type
–	vpce-0b13f04855c420629	vpc-0b06486e2412c4e09 | Endpoints-blog...	com.amazonaws.us-west-2.ec2	Interface

Figure 9: Review existing VPC endpoints

To create an interface VPC endpoint on the console, refer to the documentation Create a VPC endpoint. To create gateway endpoints for Amazon S3 and Amazon DynamoDB, refer to the documentation Create a gateway endpoint and Gateway endpoints for Amazon DynamoDB, respectively.

We encourage an infrastructure as code (IaC) approach using tools such as AWS CloudFormation or HashiCorp Terraform to create and manage VPC endpoints at scale. An IaC approach lowers the time to deployment through automation and makes deployments more repeatable and less error-prone by removing manual toil. Reference the AWS::EC2::VPCEndpoint resource for CloudFormation or aws_vpc_endpoint resource for Terraform.

Cleanup

To delete the resources provisioned, delete the associated CloudFormation stack from each step. Navigate to the CloudFormation dashboard on the console, select **Stacks**, and choose the appropriate stack name. Choose the **Delete** button to delete all resources in the stack. Here are the previously suggested stack names.

- Step 1 stack name: **vpc-endpoint-blog**
- Step 2 stack name: **vpc-flow-logs**

Conclusion

Using VPC endpoints allows for secure private routing, optimal network path, and potential cost savings. When there are many AWS services in use from multiple VPCs, it can be challenging to know which VPC endpoints to implement.

In this blog, we reviewed an approach for checking which Amazon VPC endpoints should be considered based on AWS service usage. We also detailed steps for automating this solution for a new or existing VPC. Once you have identified Amazon VPC endpoints, we recommend creating them using an infrastructure as code approach.

Review the documentation on Amazon VPC to explore AWS PrivateLink further and read about other related services.