

Contents

[1. About Test Drive 3](#_Toc478659666)

[2. What is Fortinet? 3](#_Toc478659667)

[3. Objective of Fortinet Transit VPC 3](#_Toc478659668)

[4. High Level Network Architecture 4](#_Toc478659669)

[5. Getting Started 6](#_Toc478659670)

# About Test Drive

This Test Drive enables customers to rapidly try Fortinet Next-Generation Firewall features using Amazon Web Services (AWS) cloud infrastructure services. Fortinet has several security offerings available in the AWS Marketplace with free trial, bring-your-own-license (BYOL), and hourly metering-based billing options.

# What is Fortinet?

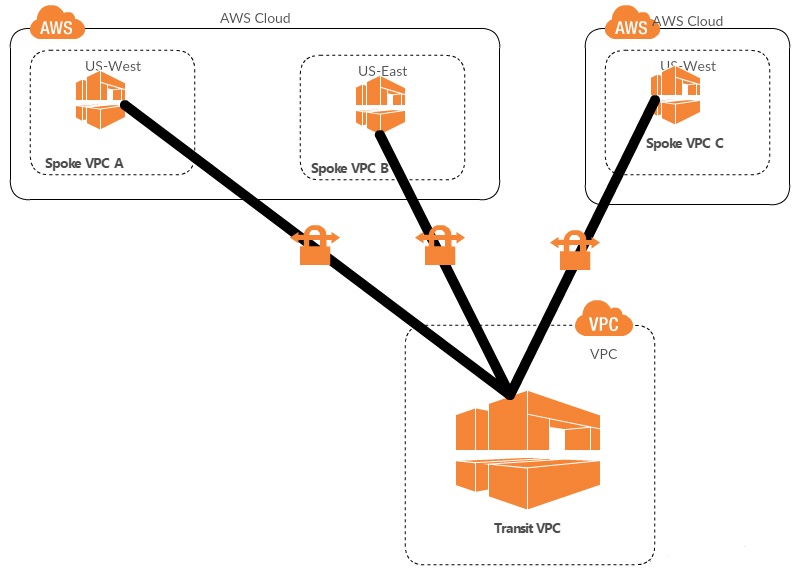
Fortinet vision has been to deliver broad, truly integrated, high-performance security across the IT infrastructure.  It develops and markets [cybersecurity](https://en.wikipedia.org/wiki/Cybersecurity) software, appliances and services, such as firewalls, anti-virus, intrusion prevention and endpoint security, among others.

Fortinet’s unique security fabric combines Security Processors, an intuitive operating system, and applied threat intelligence to give you proven security, exceptional performance, and better visibility and control-while providing easier administration.

Fortinet’s flagship enterprise firewall platform, FortiGate, is available in a wide range of sizes and form factors to fit any environment, and provides a broad array of next generation security and networking functions.

# Objective of Fortinet Transit VPC

One of the common strategy for connecting multiple, geographically disperse VPCs and remote networks is to create a transit VPC that serves as a global network transit centre.   
A transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks.



This design can save time, effort and reduce costs, as it is implemented virtually without the traditional expense of establishing a physical presence in a colocation transit hub or deploying physical network gear.

The Cloud formation template configures and runs Fortinet Gateway on AWS. The solutions below is used to display primarily FortiGate's features on AWS Cloud:

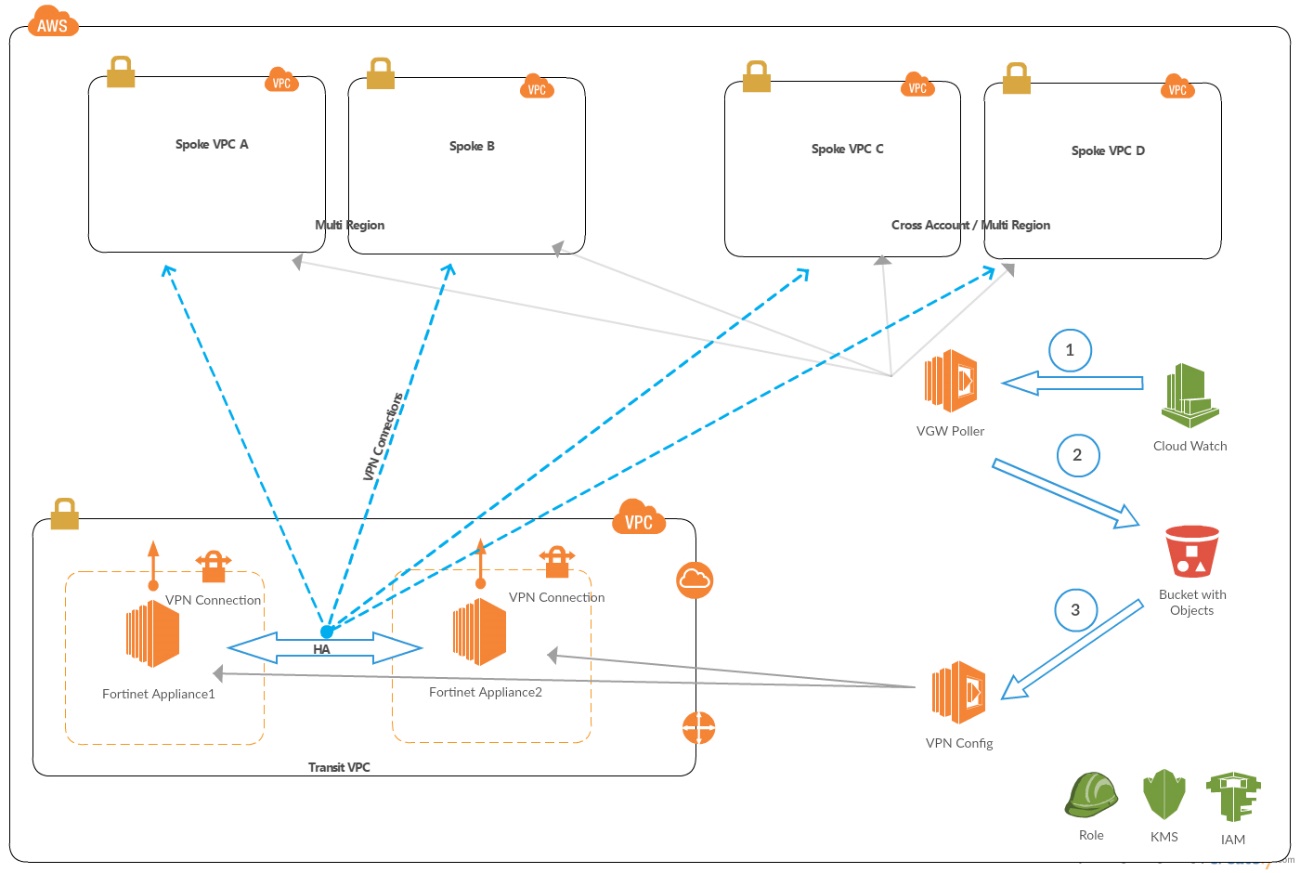
• FortiGate VPN appliance in a typical hub-and-spoke network topology to connect multiple VPC’s across Regions & Accounts.

• Automate VPN connection configuration and setup as new spoke VPC join the topology.

• Support both BYOL as well as Pay-as-you-go options.

# High Level Network Architecture

Core Components Architecture



**Note**:

The above diagram is simplified for better readability. Each FortiGate will be connected to the entire spoke VPCs independently in a redundant manner. Both FortiGate's operate in Active-Active mode.

The Core Components for this solution are:

1. AWS VPC, VGW
2. Fortinet Appliances - FortiGate (BYOL/ Pay as you Go)
3. AWS S3
4. AWS EC2
5. AWS Lambda
6. AWS Cloud Watch
7. AWS IAM Roles
8. AWS KMS

Following provides high-level solution design. Detailed design and technical documentation (in word format) will be provided as part of the deliverable.

1. The solution deploys a transit VPC

* In a user provided AWS region
* In a user provided CIDR.

1. It deploys one or more spoke VPCs.

* Spoke VPCs could be spread across multiple regions
* Spoke VPCs could be in one more AWS Accounts and are connected to transit network.
* Spoke VPCs do not overlap with each other or with transit VPC

1. This highly available design deploys two FortiGate VPN appliances into separate Availability Zones of a dedicated transit VPC.

* Each FortiGate instance has an associated Amazon CloudWatch alarm that enables automatic recovery of the instance if they underlying EC2 hardware fails.

1. Spoke VPCs are connected to the transit network through dynamically routed VPN connections between their virtual private gateways (VGWs) and the FortiGate instances.
2. The user has an option to select FortiGate BYOL or Pay as you go, if the user opts for BYOL option, the associated license file should be uploaded to accessible url endpoint (AWS S3 or others) and provided as an input to the template.
3. FortiManager optionally can be deployed in a separate subnet in the transit VPC. However, the integration to Fortinet Appliances has be performed manually.

The automated process for adding a new spoke VPC, as part of this solution, is as follows:

1. Every (1) minute, an Amazon CloudWatch event invokes the VGW Poller Lambda function, which iterates through each AWS Region of a one or more customer’s accounts, searching for appropriately tagged spoke VGWs (default tag key transitvpc:spoke, default tag value true) that do not have existing transit VPC VPN connections.
2. When the VGW Poller identifies an applicable spoke VGW, it creates the corresponding customer gateways (if required) and VPN connections to each FortiGate Appliance, and then saves this connection information to an Amazon S3 bucket using S3 SSE-KMS.   
   All data in the S3 bucket is encrypted using a solution-specific AWS KMS managed customer master key (CMK).
3. The S3 Put event invokes the VPN Configurator Lambda function, which parses the VPN connection information and generates the necessary config files to create new VPN connections.
4. The VPN Config (Lambda function) pushes the configuration to the VPN Appliance instances using SSH.
5. As soon as the VPN configuration is applied onto the FortiGate instances, the VPN tunnels come up and Border Gateway Protocol (BGP) neighbour relationships are established to the spoke VPCs.

The process for removing an existing spoke VPC is as follows:

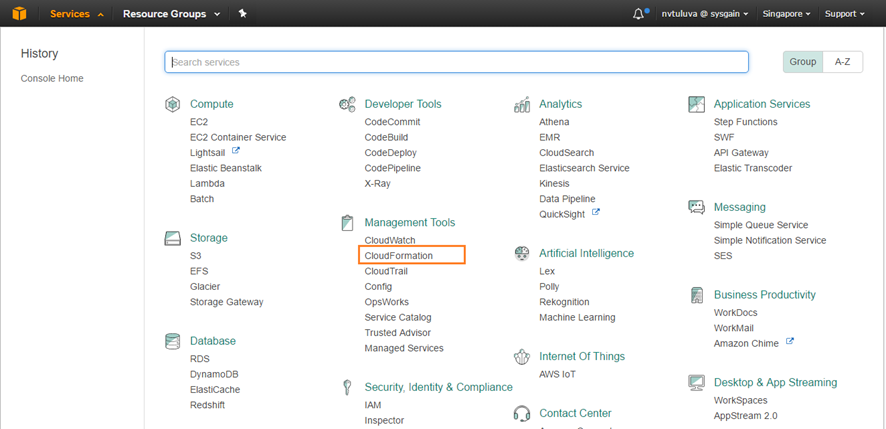
1. Every (1) minute, an Amazon CloudWatch event invokes the VGW Poller Lambda function, which iterates through each AWS Region of one or more customer’s accounts, searching for spoke VGWs to remove.

Spoke VGWs to remove have existing transit VPC connections, but either lack the spoke VPC tag (because it has been removed) or the spoke VPC tag value has been changed from the expected value.

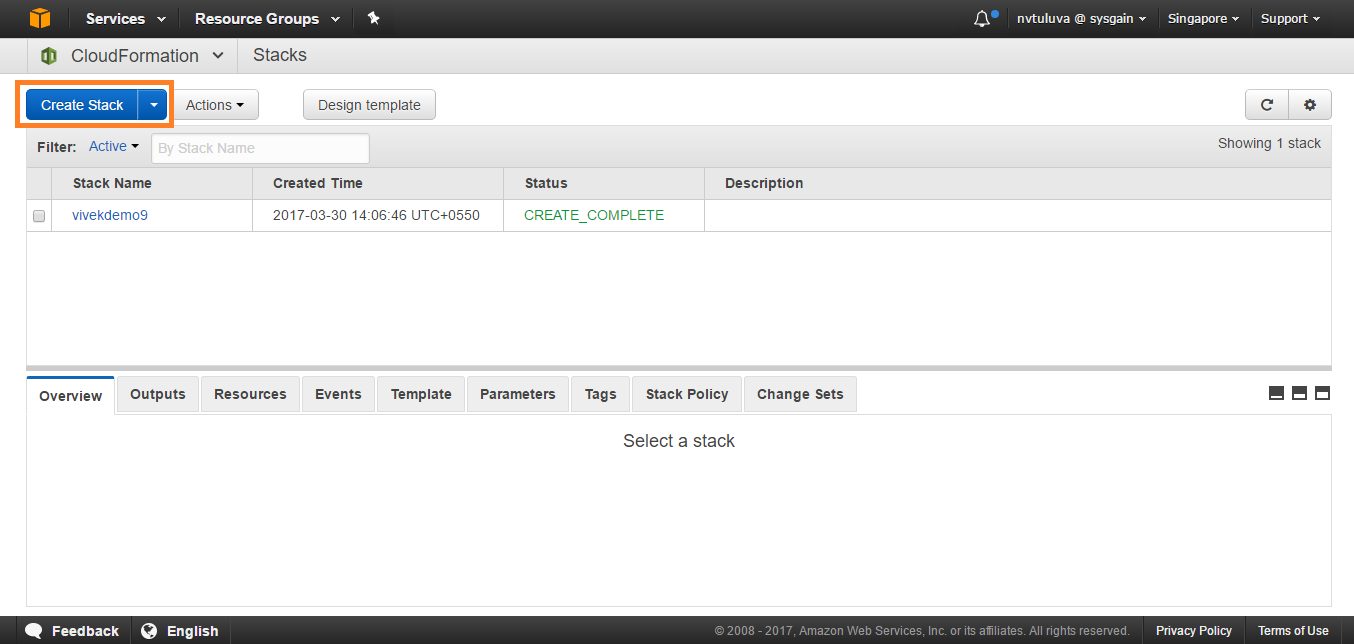
1. When the VGW Poller identifies an applicable spoke VGW for removal, it deletes existing VPN connections to each FortiGate, saves the existing connection information to an Amazon S3 bucket using S3 SSE-KMS, and deletes any corresponding customer gateways (if no longer required).
2. The S3 Put event invokes the VPN Configurator Lambda function, which parses the VPN connection information and generates the necessary config files to remove the VPN connections.
3. The VPN Config pushes the connection-removal configuration to the FortiGate instances using SSH.

# Getting Started

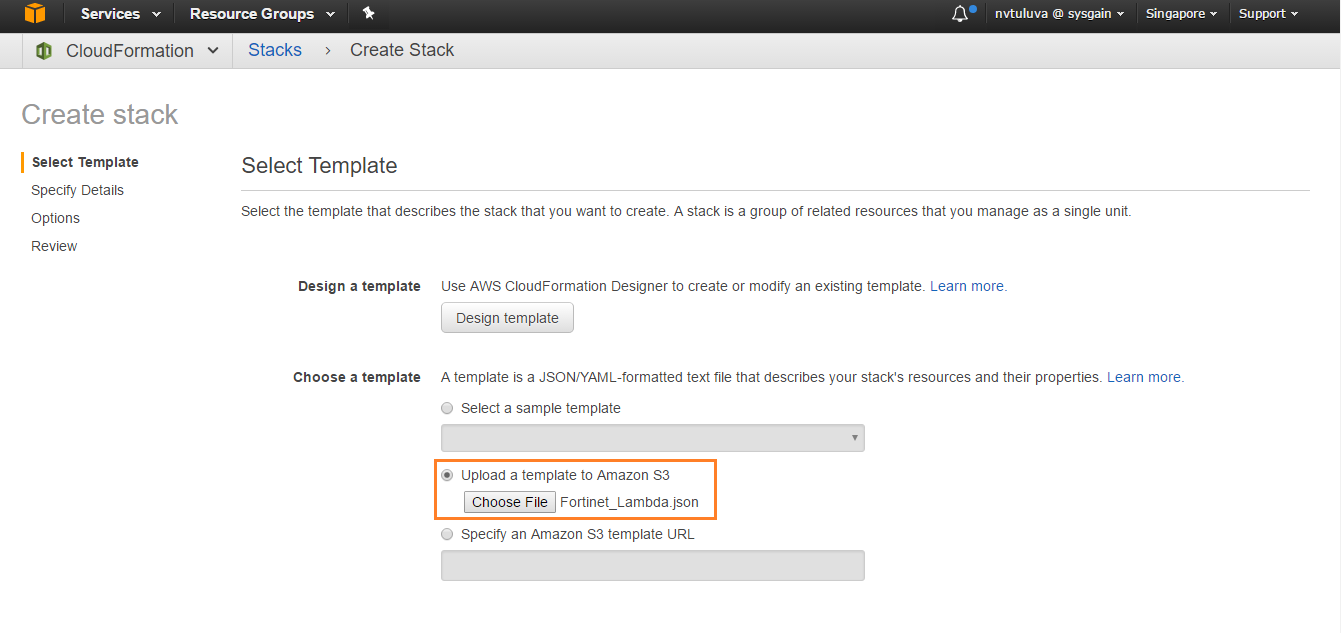
1. Open AWS console with your credentials and select the deployment location.
2. Click on **Services** and navigate to **CloudFormation** under **Management Tools.**



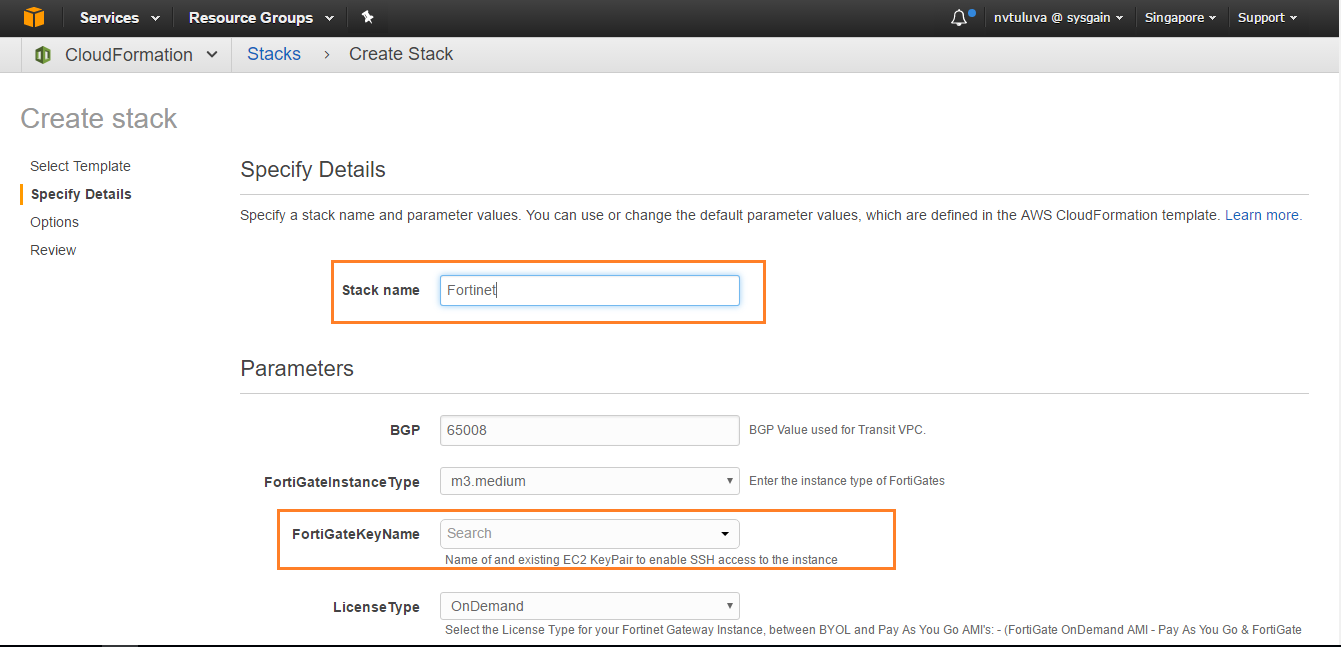
1. Click on create task.



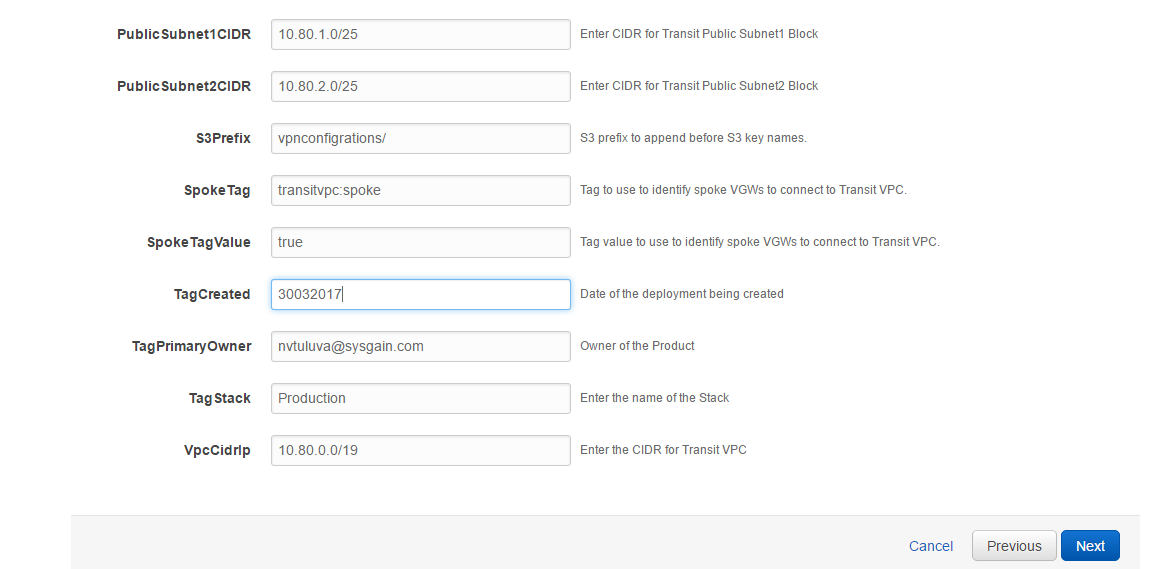
1. In **Select Template** page Choose to upload a template to Amazon S3 and click on **Next**.



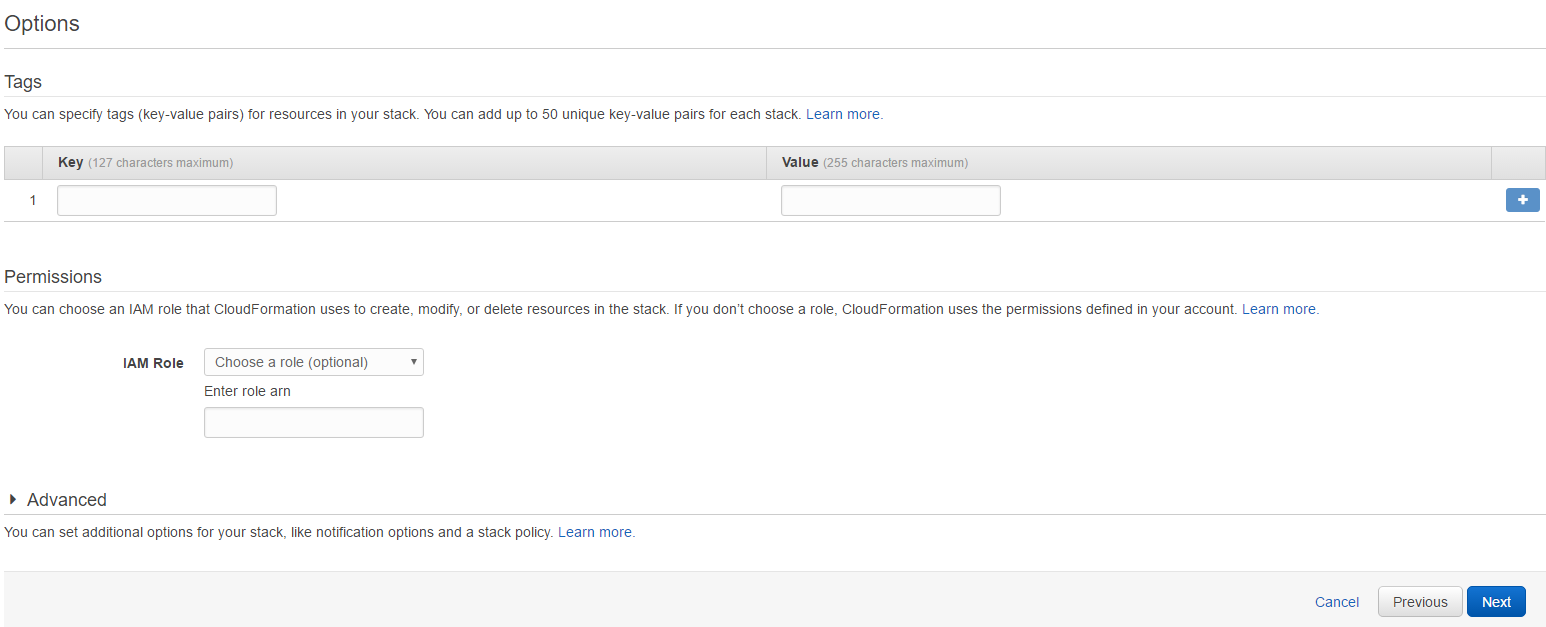
1. A Specify Details page opens. Enter the **Stack name** and give the **FortiGateKeyName.**



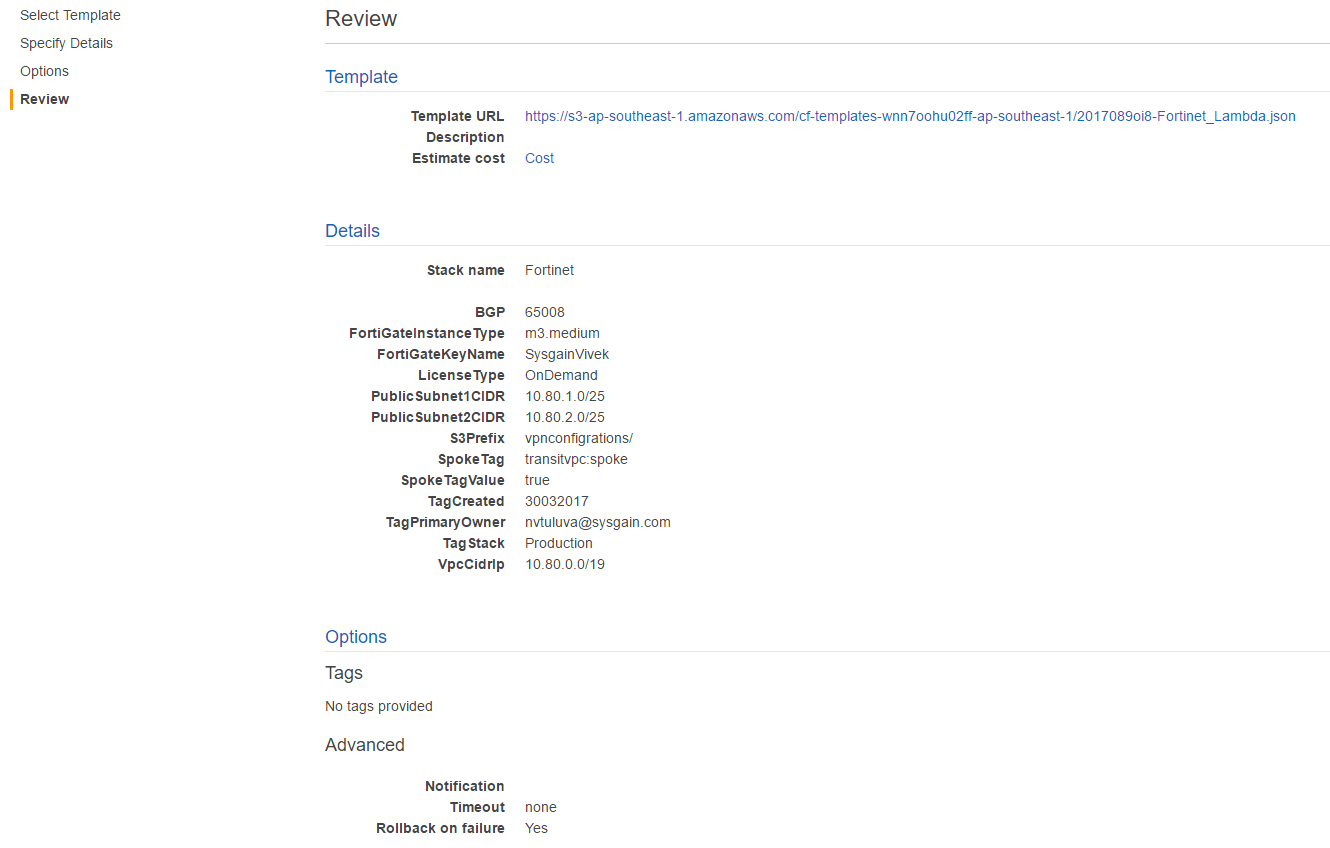
1. Enter the Date of deployment being created in **TagCreated** section of same page and click on **Next**, which lead to Options page.



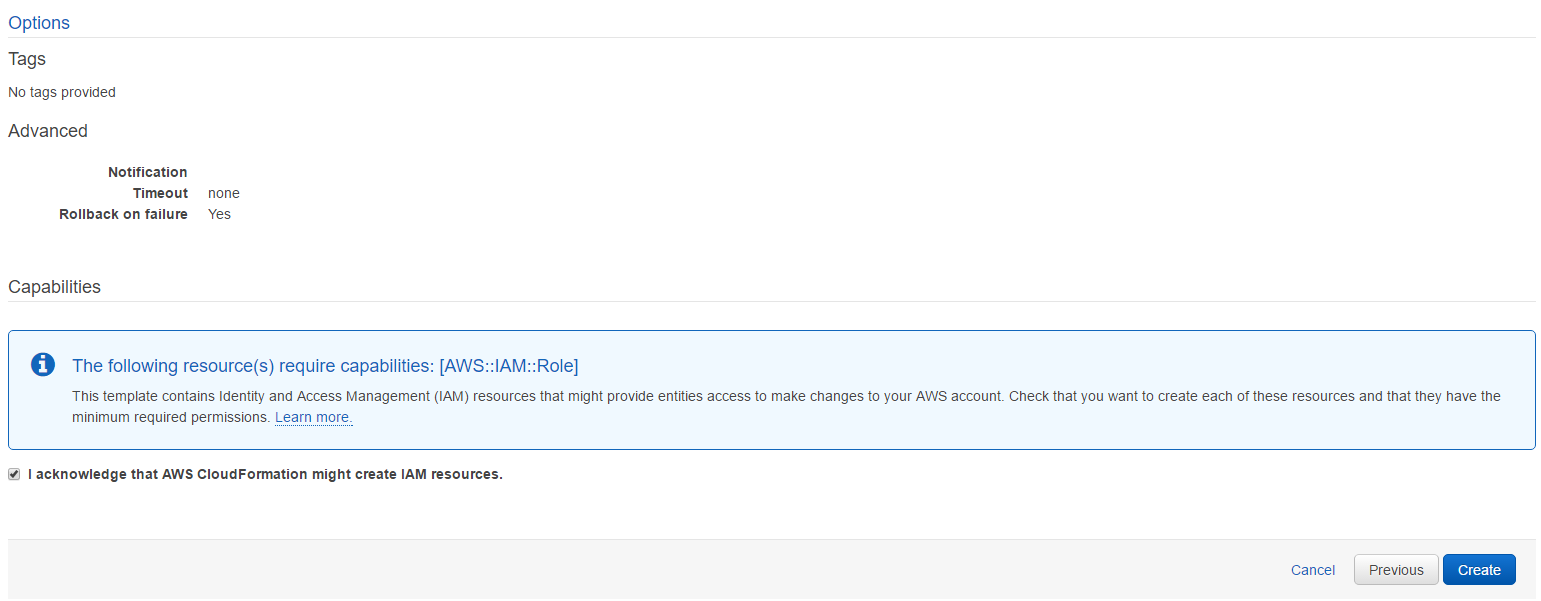
1. In this page, enter the Tags giving values for **Key** section and **Value** section and click on **Next**.



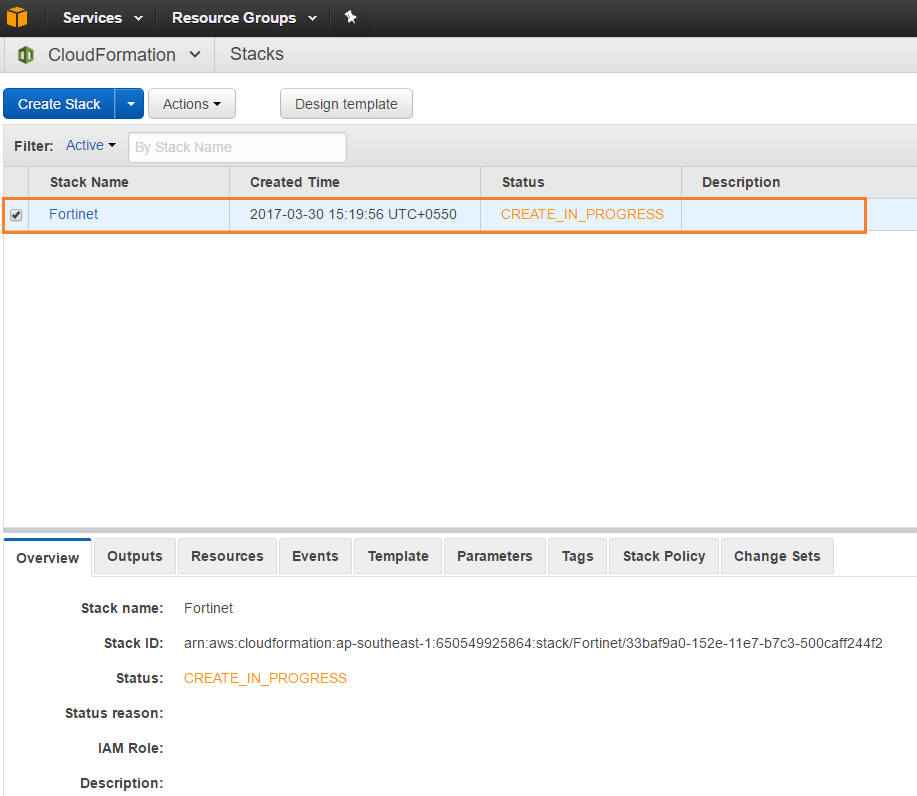
1. A Review pages opens which gives the overview of the details entered.

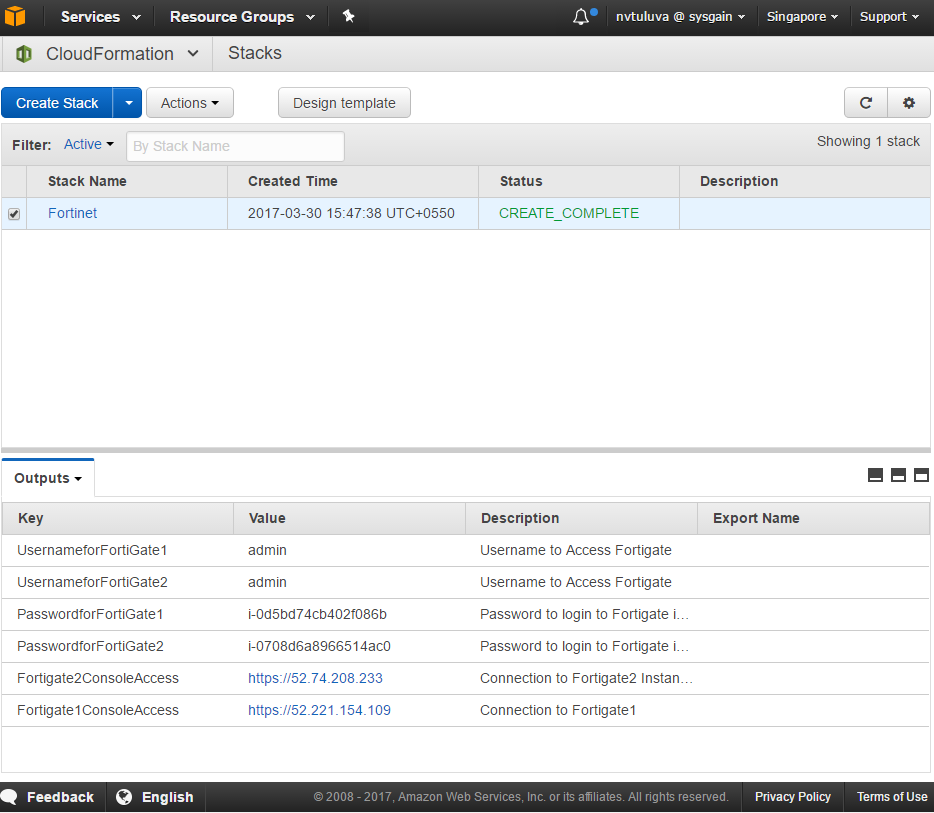


1. Check the box of ***I acknowledge that AWS CloudFormation might create IAM resources*** and click on **Create**.

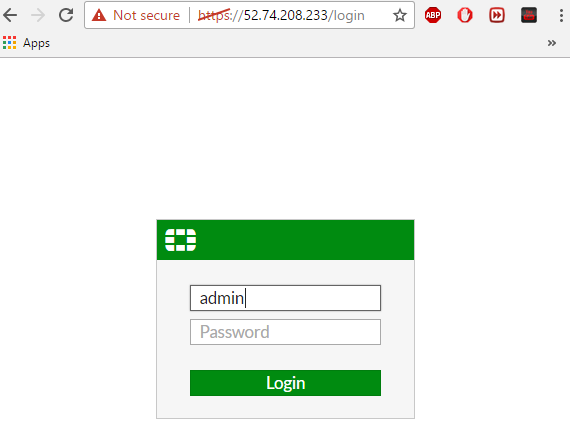


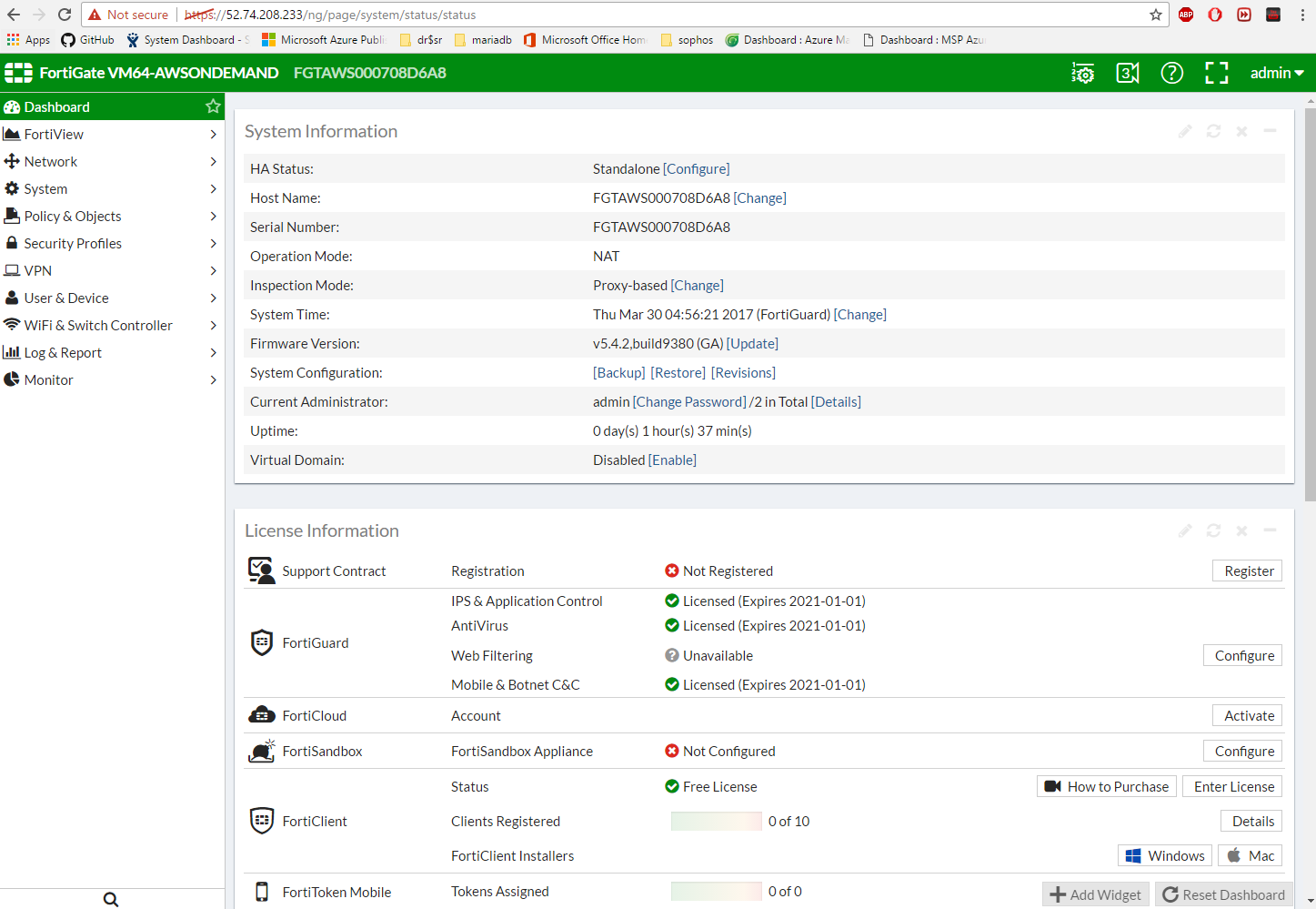
1. This leads to stack creation page showing the progress of the stack created. Click on the output sections below to view the outputs.





1. We can access the Fortigate console by clicking on the value and giving the **username** as ***admin*** and **password** from the output section relevant to the console.





1. Navigate to **Services>INSTANCES>Instances** to check for launched instances.

