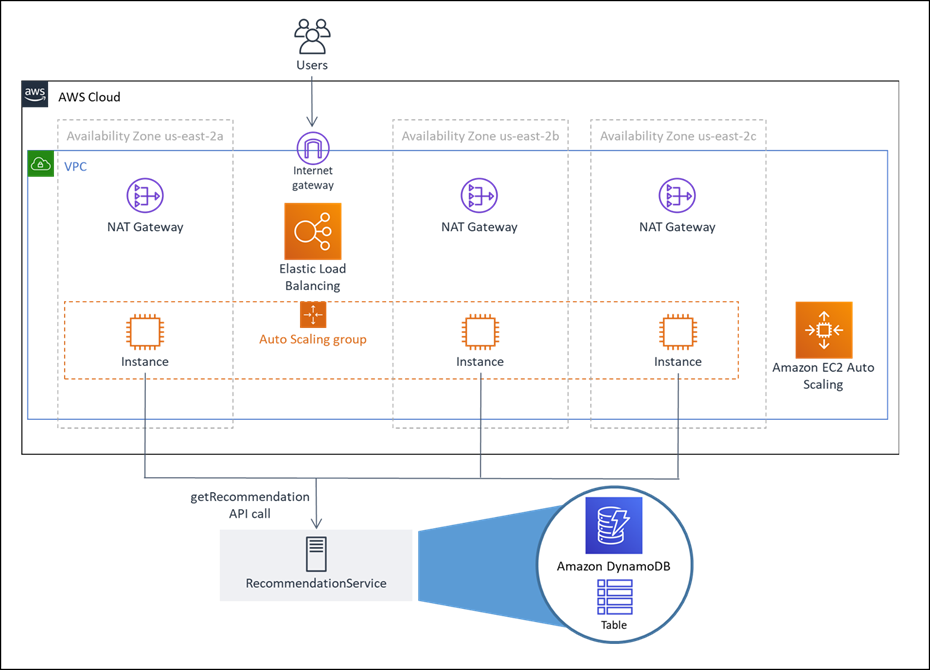
# DEPLOY A RELIABLE MULTI-TIER INFRASTRUCTURE USING CLOUDFORMATION

## Introduction

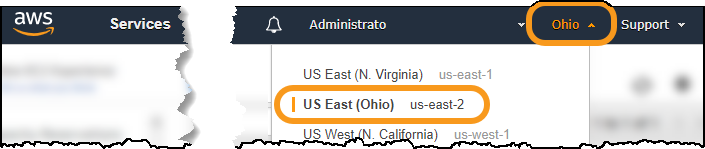
This hands-on lab will guide you through the steps to improve reliability of a service by using automation to deploy a reliable cloud infrastructure. When this lab is completed, you will have deployed two CloudFormation templates. The first will deploy an Amazon Virtual Private Cloud (VPC). The second will deploy into your VPC, a reliable 3-tier infrastructure using Amazon EC2 distributed across three Availability Zones. You will then review the features of the deployed infrastructure and learn how they contribute to reliability.

The architecture of the infrastructure you will deploy is represented by this diagram:



### **1.1 Log into the AWS console**

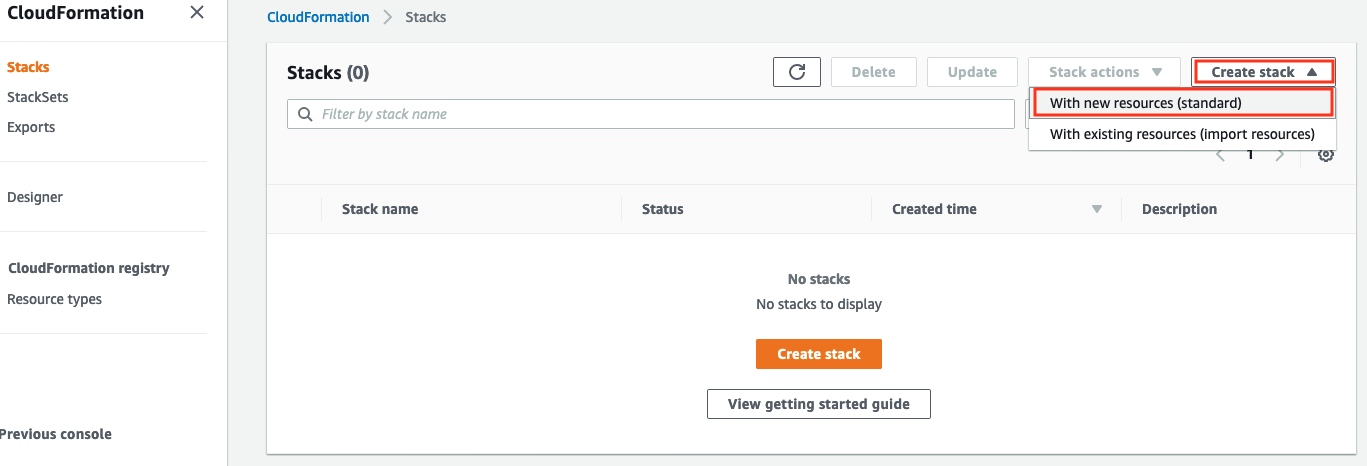
### **1.2 Configure your AWS Region**

1. Select the **Ohio** region. This region is also known as **us-east-2**, which you will see referenced throughout this lab.[](https://wellarchitectedlabs.com/Reliability/100_Deploy_CloudFormation/Images/SelectOhio.png)
   * AWS offers you the ability to deploy to over 20 regions located across the globe
   * Each region is fully isolated from the others to isolate any issues and achieve high availability,
   * Each region is comprised of multiple Availability Zones, which are fully isolated partitions of our infrastructure (more on this later)

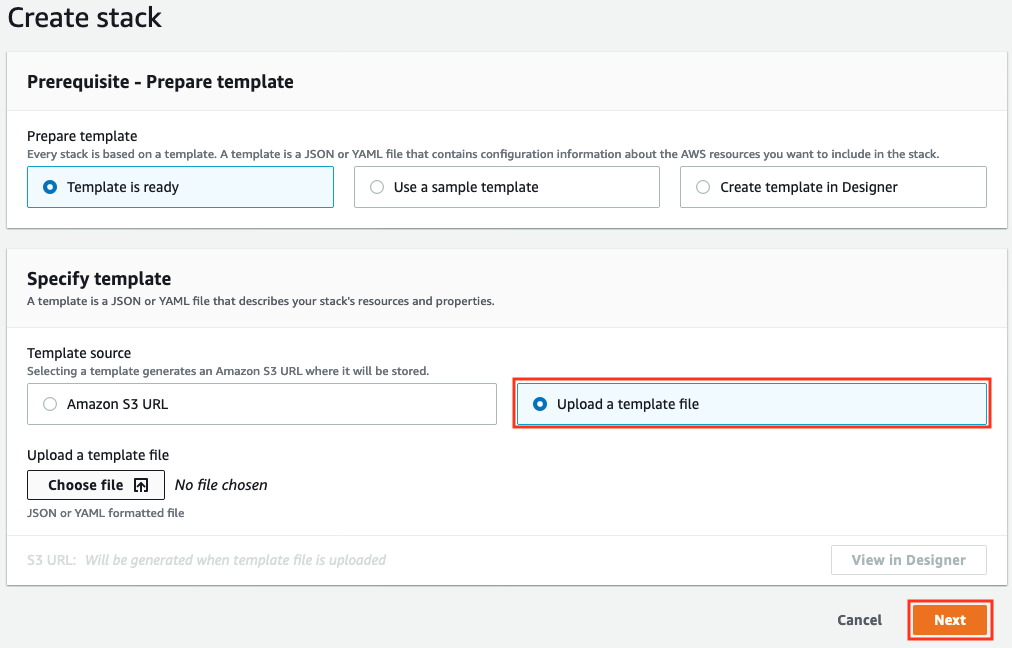
#### **1.3 Deploy the VPC infrastructure**

This step will create the VPC and all components using the example CloudFormation template.

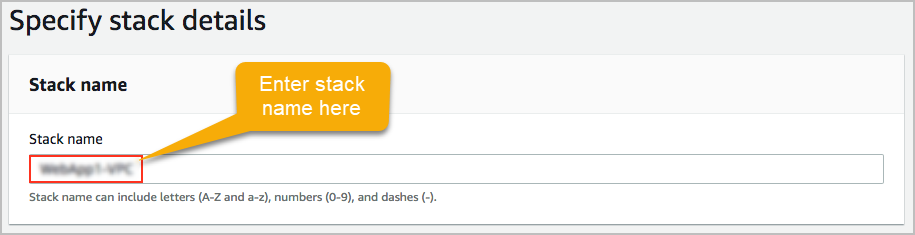
1. Download the latest version of the CloudFormation template from Github – vpc-alb-app-db.yaml
2. Sign in to the AWS Management Console, select your preferred region, and open the CloudFormation console at <https://console.aws.amazon.com/cloudformation/> .
3. Click **Create Stack**, then **With new resources (standard)**.

[](https://wellarchitectedlabs.com/Common/Create_VPC_Stack/Images/cloudformation-createstack-1.png)

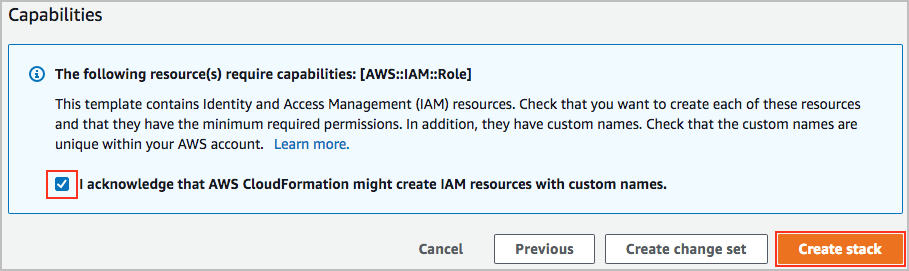
1. Click **Upload a template file** and then click **Choose file**.

[](https://wellarchitectedlabs.com/Common/Create_VPC_Stack/Images/cloudformation-createstack-2.png)

1. Choose the CloudFormation template you downloaded in step 1, return to the CloudFormation console page and click **Next**.
2. Enter the following details:
   * **Stack name**: The name of this stack. For this lab, use **WebApp1-VPC** and match the case.
   * **Parameters**: Parameters may be left as defaults, you can find out more in the description for each.

[](https://wellarchitectedlabs.com/Common/Create_VPC_Stack/Images/cloudformation-vpc-params.png)

1. At the bottom of the page click **Next**.
2. In this lab, we use tags, which are key-value pairs, that can help you identify your stacks. Enter Owner in the left column which is the key, and your email address in the right column which is the value. We will not use additional permissions or advanced options so click **Next**.
3. Review the information for the stack. When you’re satisfied with the configuration, at the bottom of the page check **I acknowledge that AWS CloudFormation might create IAM resources with custom names** then click **Create stack**.

[](https://wellarchitectedlabs.com/Common/Create_VPC_Stack/Images/cloudformation-vpc-createstack-final.png)

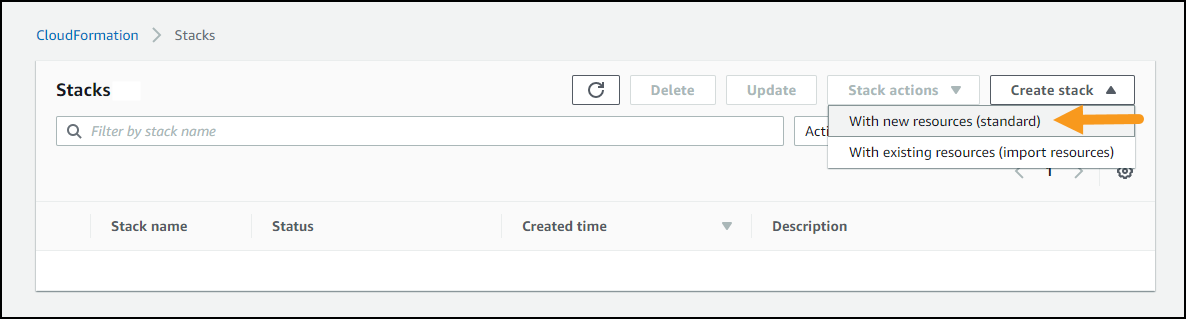
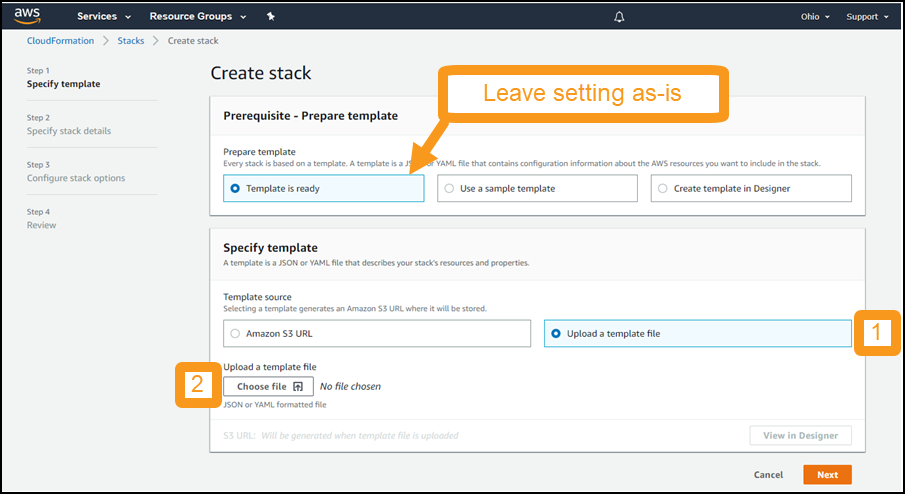
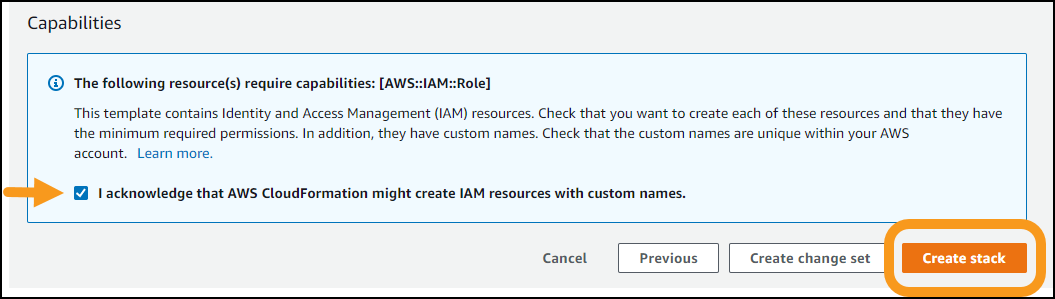
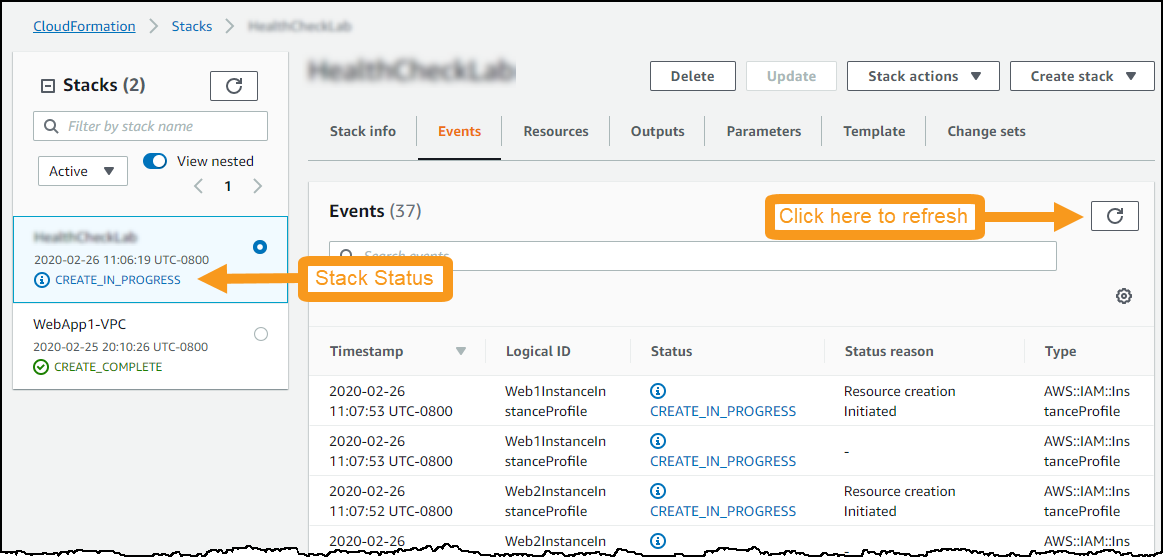
1. After a few minutes the final stack status should change from CREATE\_IN\_PROGRESS to CREATE\_COMPLETE. You can click the **refresh** button to check on the current status. You have now created the VPC stack (well actually CloudFormation did it for you).
2. When the stack status is CREATE\_COMPLETE, you can continue to the next step.

# DEPLOY WEB APPLICATION AND INFRASTRUCTURE USING CLOUDFORMATION

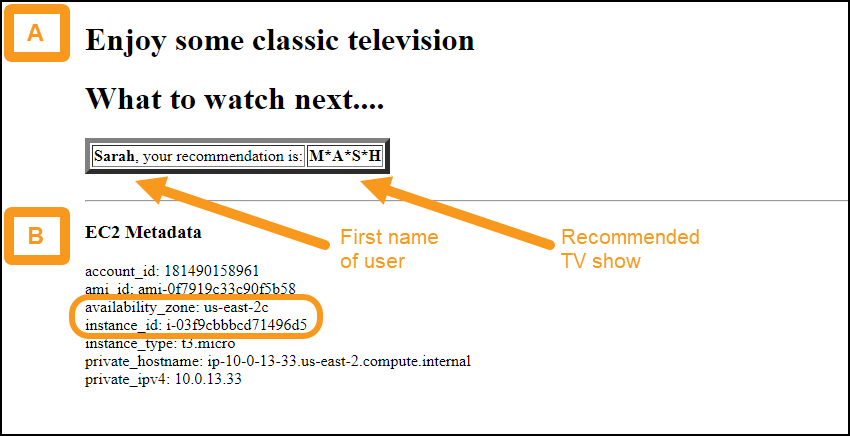
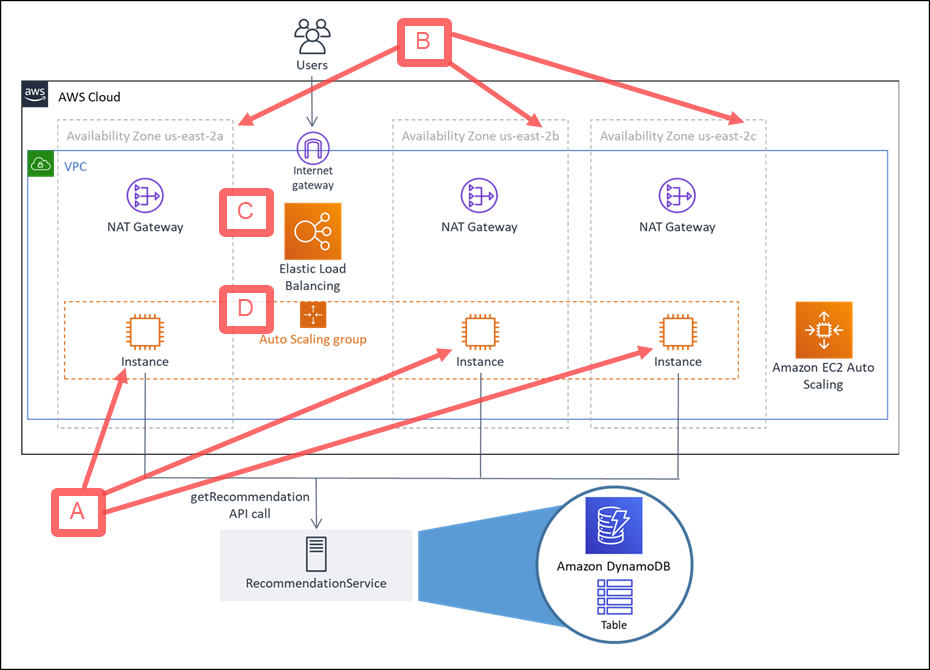
|  |  |
| --- | --- |
| Previous Step | Next Step |

Wait until the VPC CloudFormation stack **status** is CREATE\_COMPLETE, then continue. This will take about four minutes.

* Download the CloudFormation template : staticwebapp.yaml

1. Go to the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation> and click **Create Stack** > **With new resources**[](https://wellarchitectedlabs.com/Common/images/CreateNewCloudFormationStack/CFNCreateStackButton.png)
2. Leave **Prepare template** setting as-is
   * For **Template source** select **Upload a template file**
   * Click **Choose file** and supply the CloudFormation template you downloaded: staticwebapp.yaml[](https://wellarchitectedlabs.com/Common/images/CreateNewCloudFormationStack/CFNUploadTemplateFile.png)
3. Click **Next**
4. For **Stack name** use **CloudFormationLab**
5. **Parameters**
   * Look over the Parameters and their default values.
   * Click **Next**
6. For **Configure stack options** we recommend configuring tags, which are key-value pairs, that can help you identify your stacks and the resources they create. For example, enter Owner in the left column which is the key, and your email address in the right column which is the value. We will not use additional permissions or advanced options so click **Next**.
7. For **Review**
   * Review the contents of the page
   * At the bottom of the page, select **I acknowledge that AWS CloudFormation might create IAM resources with custom names**
   * Click **Create stack**[](https://wellarchitectedlabs.com/Common/images/CreateNewCloudFormationStack/CFNIamCapabilities.png)
8. This will take you to the CloudFormation stack status page, showing the stack creation in progress.
   * Click on the **Events** tab
   * Scroll through the listing. It shows the activities performed by CloudFormation (newest events at top), such as starting to create a resource and then completing the resource creation.
   * Any errors encountered during the creation of the stack will be listed in this tab.[](https://wellarchitectedlabs.com/Common/images/CreateNewCloudFormationStack/CFNStackInProgress.png)
9. When it shows **status** CREATE\_COMPLETE, then you are finished with this step.

# EXPLORE THE WEB APPLICATION

1. Go to the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>.
   * Wait until **CloudFormationLab** stack **status** is CREATE\_COMPLETE before proceeding. This should take about four minutes
   * Click on the **CloudFormationLab** stack
   * Click on the **Outputs** tab
   * For the Key **WebsiteURL** copy the value. This is the URL of your test web service
     + Hint: it will start with *http://healt-alb* and end in *<aws region>.elb.amazonaws.com*
2. Click the URL and it will bring up the website:
   * Troubleshooting: if you see an error such as 502 Bad Gateway, then wait 60 seconds and try again. It takes some time for the servers to initialize.[](https://wellarchitectedlabs.com/Reliability/300_Health_Checks_and_Dependencies/Images/DemoWebsite.png)
3. The website simulates a recommendation engine making personalized suggestions for classic television shows. You should note the following features:
   * Area A shows the personalized recommendation
     + It shows first name of the user and the show that was recommended
     + The workshop simulation is simple. On every request it chooses a user at random, and shows a recommendation statically mapped to that user. The user names, television show names, and this mapping are in a DynamoDB table, which is simulating the **RecommendationService**
   * Area B shows metadata which is useful to you during the lab
     + The **instance\_id** and **availability\_zone** enable you to see which EC2 server and Availability Zone were used for each request
4. Use the following architectural diagram as you explore the site[](https://wellarchitectedlabs.com/Reliability/100_Deploy_CloudFormation/Images/ArchitectureOverviewAnnotated.png)
   * **A** - There is one EC2 instance deployed per Availability Zone
   * **B** - Refresh the website several times, note that the EC2 instance and Availability Zone change from among the three available
   * **C** - Elastic Load Balancing (ELB) is used here. An Application Load Balancer receives each request and distributes it among the available EC2 server instances across Availability Zones.
     + The requests are stateless, and therefore can be routed to any of the available EC2 instances
   * **D** - The EC2 instances are in an [Amazon EC2 Auto Scaling Group](http://aws.amazon.com/ec2/autoscaling) . This Auto Scaling Group was configured to maintain three instances, therefore if one instance is detected as unhealthy it will be replaced to maintain three healthy instances.
     + AWS Auto Scaling can also be configured to scale up/down dynamically in response to workload conditions such as CPU utilization or request count.