# Setting up a VPC MarkLogic Cluster using AWS Cloud Formation

The purpose of this document is to detail how AWS Cloud Formation can be used to set up a MarkLogic Cluster in an AWS[[1]](#footnote-1) VPC[[2]](#footnote-2) environment. It should be read in conjunction with two supplied files – MarkLogic-Sample-VPC-1.json and ThreePlusClusterForVPC-1.json. If reading this via GitHub, these files are in the same directory as this document.

# Background

MarkLogic includes an in-built managed cluster capability allowing it to be both deployed easily to AWS and to take advantage of the high availability features that AWS offers. For full coverage of MarkLogic’s AWS capabilities see the online EC2 guide - <http://docs.marklogic.com/guide/ec2.pdf>.

The recommended mechanism for deploying MarkLogic is using Cloud Formation Templates (CFTs). These allow deterministic repeatable deployment and also simplify the process of expanding or upgrading the cluster.

A standard CFT for MarkLogic deployment is available[[3]](#footnote-3) but for deployment to an AWS Virtual Private Cloud (VPC) a customized template is required as it is not possibly to craft a generic template that would allow for all customer requirements and configurations.

The purpose of this document and associated assets is to provide a sample CFT for MarkLogic setup in a VPC and to show how to use it. It is hoped that this will be instructive.

# Sample VPC

In order to demonstrate deployment into a VPC, it is necessary to have a VPC to deploy into. The accompanying template, MarkLogic-Sample-VPC-1.json can be used to create such a VPC.

This template will create

* Externally ( to the VPC ) accessible sub-nets in three availability zones. Use of three availability zones is required to make full use of MarkLogic’s resiliency features
* Three internal sub-nets – hosts placed in these will not be accessible by external resources
* Security Groups
  + rMgmtSecurityGroup - allows ssh and mstsc access from a named IP Range
  + rExternalSecurityGroup – allows http and https access from external addresses
  + rWebSecurityGroup – allows ssh access from hosts in rMgmtSecurityGroup and http/https access from hosts in rExternalSecurityGroup
  + rInternalSecurityGroup – allows ssh access from rMgmtSecurityGroup and full access from rWebSecurityGroup
* A linux ‘jump box’ placed in rMgmtSecurityGroup allowing **ssh** accesss to rInternalSecurityGroup
* A windows ‘jump box’ placed in rWebSecurityGroup allowing **web** access to rInternalSecurityGroup.
* Access control lists (ACLs). These are placeholders as they permit all traffic – the security groups control access for us. These are required however. In production these settings would normally be reviewed.
* NAT Gateways[[4]](#footnote-4) allowing hosts in the private sub-net to access the internet

# Pre-requisites

In order to deploy the cluster two pre-requisites need to be in place. There should be no difficulty with these, but for the record they are

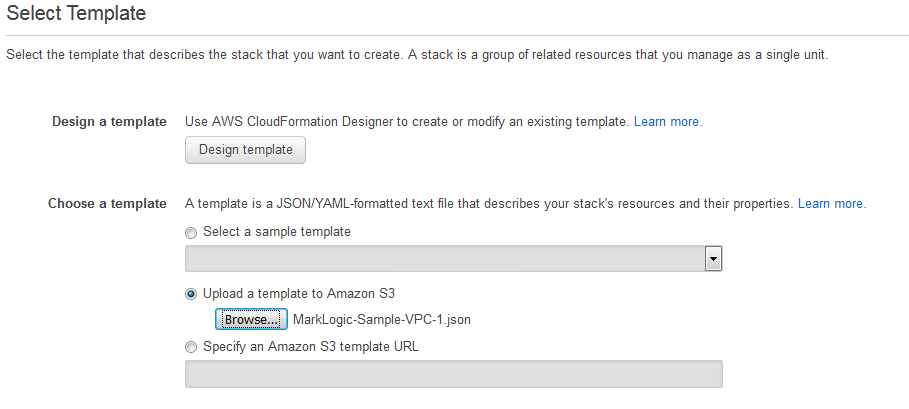
1) An AWS account – see <http://docs.marklogic.com/guide/ec2/GettingStarted#id_52961>

2) A SSH ‘key pair’ – see <http://docs.marklogic.com/guide/ec2/GettingStarted#id_24571>. This is used to secure access to instances.

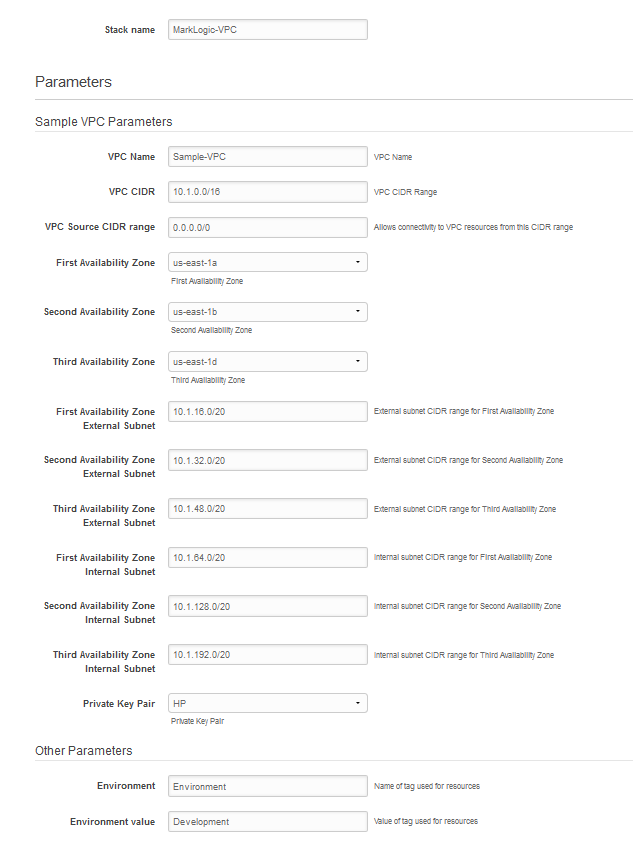
# Using the Sample VPC Template

To create the VPC we will use Amazon’s management console – <http://console.aws.amazon.com>, although tools such as the AWS Command Line Interface[[5]](#footnote-5) could also be used.

Access the Cloud Formation console via <http://console.aws.amazon.com/cloudformation>. From there, choose ‘Create Stack’. In the next page, choose ‘upload template to S3’ then browse to a location where the Sample VPC template has been saved as per the screenshot below. Select the required template then click the ‘next’ button.



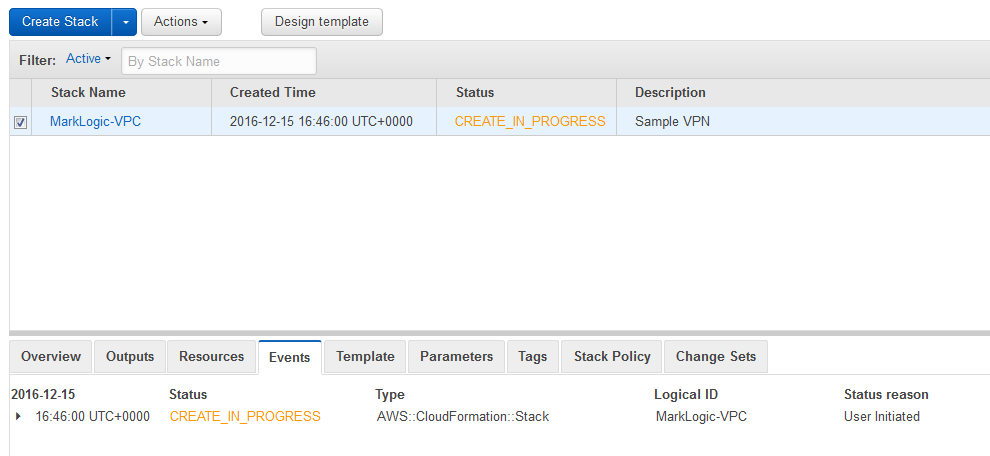
The next screen requires specification of cluster parameters.



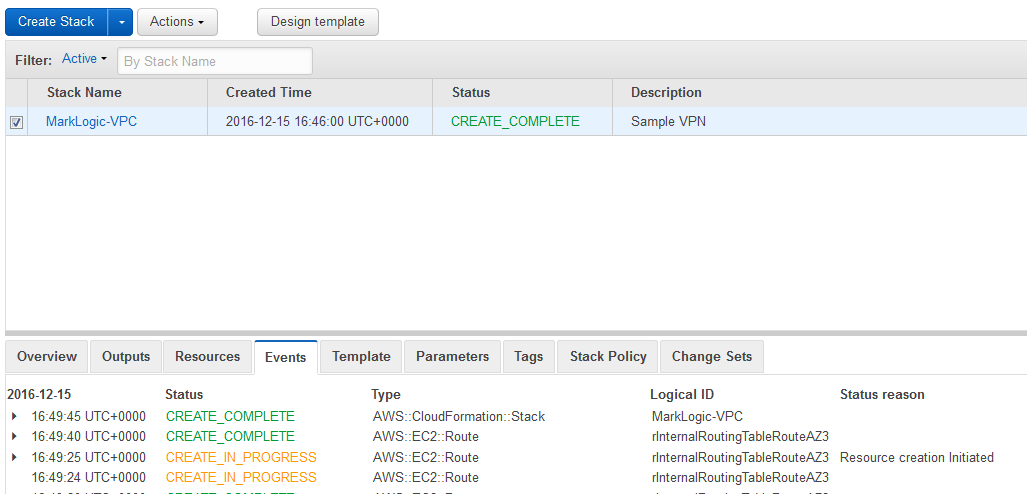
Full detail of parameters is given below but for instruction purposes the default values may be accepted, with the exception of the private key pair, which will be specific to you.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **Comment** |
| pVpcName | VPC Name. Default : Sample-VPC |  |
| pVpcCidr | Internal IP address range for VPC | Defaults to 10.1.0.0/16 |
| pVpcSourceCidr | IP address range allowed management access | Can be set to 0.0.0.0/0 for testing purposes, or your own IP /32 |
| pAvailabilityZone1 | Availability zone to be used for subnet 1 | Default : us-east-1a |
| pAvailabilityZone2 | Availability zone to be used for subnet 2 | Default : us-east-1b |
| pAvailabilityZone3 | Availability zone to be used for subnet 3 | Default : us-east-1d |
| pExternalSubnetAZ1Cidr | IP range for external subnet #1 | Default : 10.1.16.0/20 |
| pExternalSubnetAZ2Cidr | IP range for external subnet #2 | Default : 10.1.32.0/20 |
| pExternalSubnetAZ3Cidr | IP range for external subnet #3 | Default : 10.1.48.0/20 |
| pInternalSubnetAZ1Cidr | IP range for internal subnet #1 | Default : 10.1.64.0/20 |
| pInternalSubnetAZ2Cidr | IP range for internal subnet #2 | Default : 10.1.128.0/20 |
| pInternalSubnetAZ3Cidr | IP range for internal subnet #3 | Default : 10.1.192.0/20 |
| pPrivateKeyPair | SSH key pair for SSH access. See section 3 | **Required** |
| pResourceTagName | Name of any additional tag required | Can leave blank |
| pResourceTagValue | Value of additional tag | Can leave blank |

After configuring, click the ‘Next’ button twice followed by ‘Create’. You will see a screen similar to what is below.



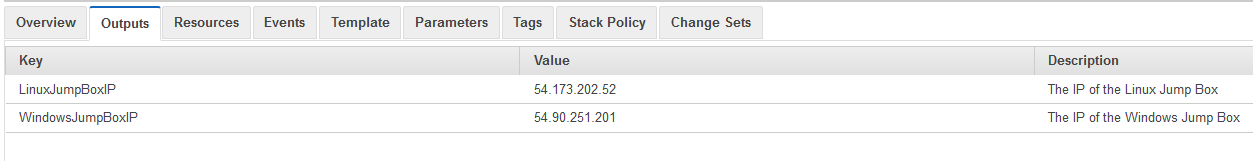
After around five minutes the stack should finish creating. Your console view will resemble



# Checking your Jump Boxes

We will be positioning the MarkLogic stack in the internal sub-nets, so we need jump boxes[[6]](#footnote-6) to access them. As above, this template sets these up for you.

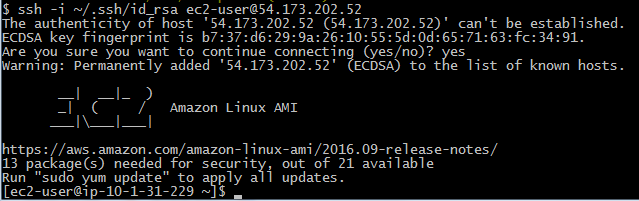
To check they are available, click the ‘Outputs’ tab in the ‘Create Stack’ view above. You should see something like the screenshot below.



## Checking Linux Jump Box

Here, my Linux jump box has IP 54.173.202.52. The private half of my ssh key pair is in .ssh/id\_rsa in my home directory. I verify my jump box is available by successfully executing

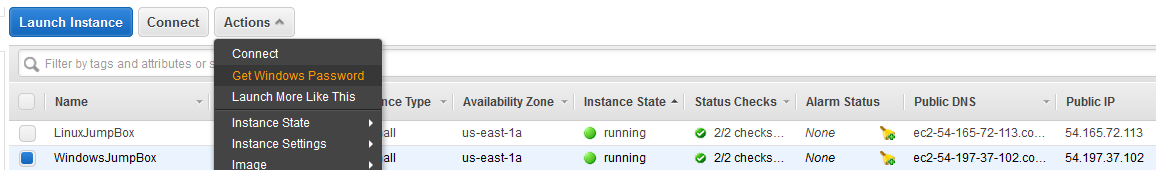
ssh -i ~/.ssh/id\_rsa ec2-user@54.173.202.52



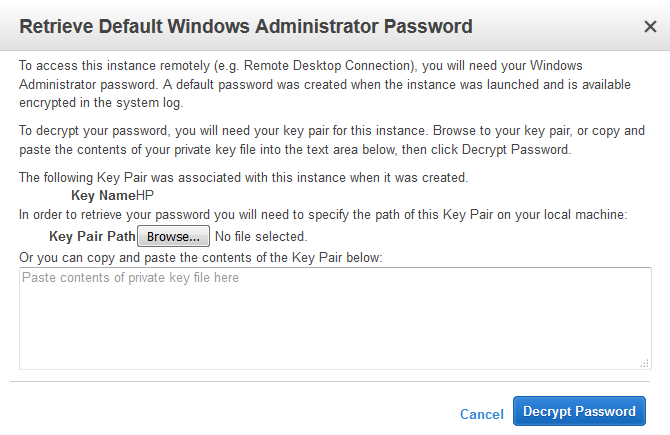
You should similarly verify for your IP/key combination. You should use user ec2-user.

## Checking Windows Jump Box

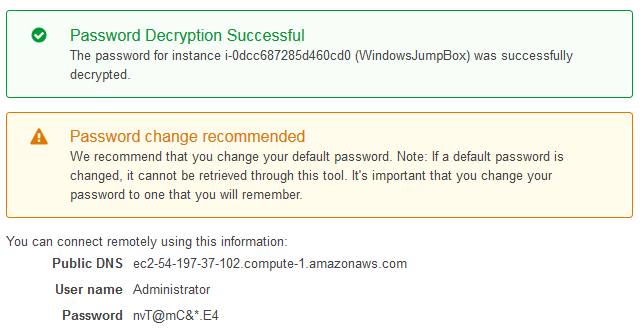
For the Windows jump box, it’s slightly more complicated, but not much.



First, because you can’t ssh into a Windows machine, you need to obtain the Windows admin password. Do this from the EC2 console[[7]](#footnote-7). Select the host with name ‘WindowsJumpBox’ and then ‘Get Windows Password’ as shown. You’ll see



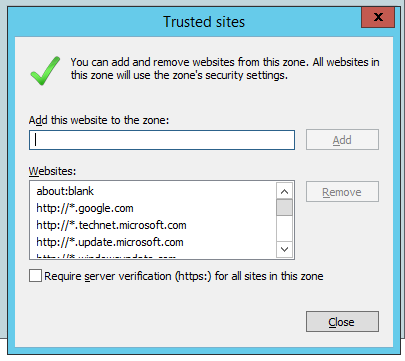
Click browse, and then select the private part of your ssh key pair[[8]](#footnote-8), then ‘Decrypt Password’. You’ll see a screen similar to



Use an RDP client (e.g. mstsc) to log into the above box using account ‘Administrator’ and the password given. You should be able to log in successfully. While you’re there, follow the recommendation above and change your password. It’s CTRL-ALT-END under Windows to do this.

To avoid difficulties with Internet Explorer, which will require explicit exemptions for any websites accessed, including the MarkLogic Elastic Load Balancer, installing Chrome[[9]](#footnote-9) is recommended. Open Internet Explorer and access [www.google.com](http://www.google.com). You’ll be told it’s untrusted. Add [http://\*.google.com](http://*.google.com) and [https://\*.google.com](https://*.google.com) as trusted sites then navigate to

<https://www.google.com/chrome/browser/desktop/index.html>



Download, install and make Chrome your default browser.

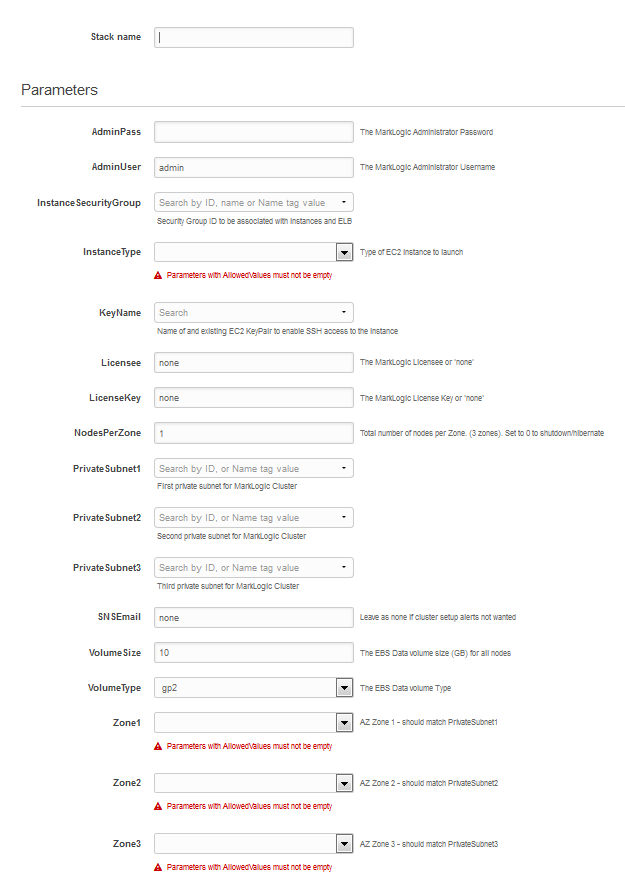
# Setting up MarkLogic in the VPC

In this section we will use the ThreePlusClusterForVPC-1.json template to set up a MarkLogic cluster.

There is not a single fixed way in which you might do this. Here we deploy MarkLogic hosts into our private subnets. We will be assigning our hosts to the rInternalSecurityGroup created above, which means in turn that they can be seen by hosts in the rWebSecurityGroup. A typical configuration might be to place the middle tier (e.g. node.js/tomcat/apache) in rWebSecurityGroup and then add in a load balancer for these assets in the rExternalSecurityGroup, which would be visible externally. Later we see how we can demonstrate that the arrangement works using our Windows jump box.

## Template Use

As previously, access the Cloud Formation console via <http://console.aws.amazon.com/cloudformation>. From there, choose ‘Create Stack’. In the next page, choose ‘upload template to S3’ then browse to a location where the MarkLogic cluster template ThreePlusClusterForVPC-1.json has been saved. Select the required template then click the ‘next’ button. You will see something like the screen below.

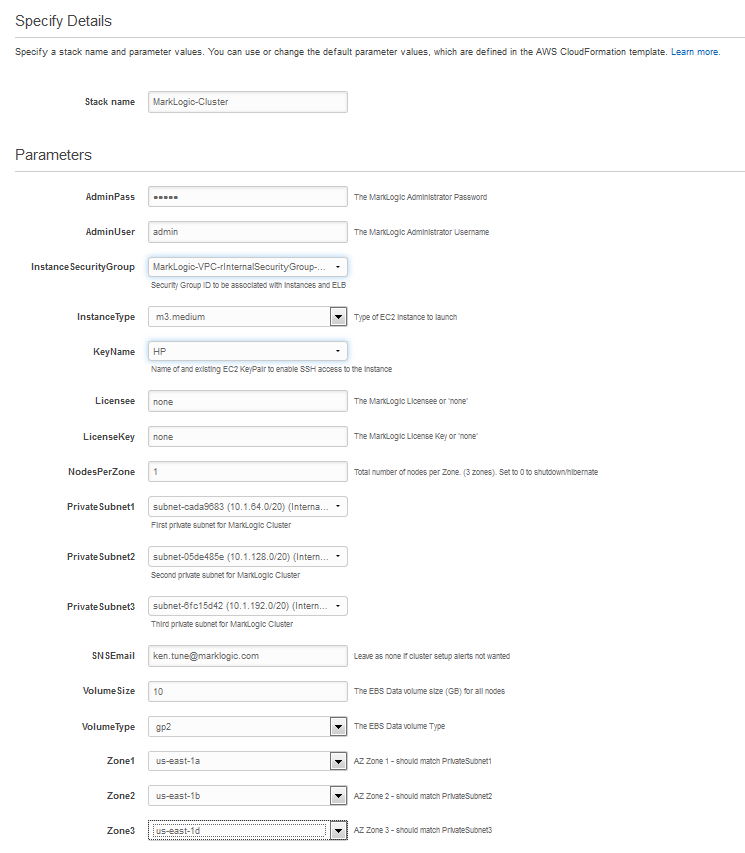


Fields are required as follows

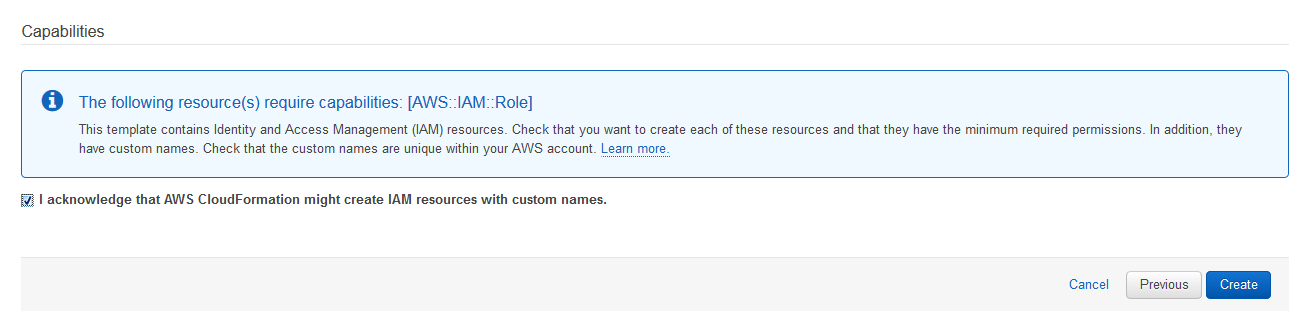
|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **Value if using MarkLogic VPC** |
| Stack Name | Name of stack – free text field |  |
| AdminPass | Configure admin user password |  |
| AdminUser | Configure admin user name |  |
| InstanceSecurityGroup | Security group for ML instances/ELB | rInternalSecurityGroup |
| InstanceType | EC2 instance type | m3.medium for instruction[[10]](#footnote-10) |
| KeyName | SSH Key pair for ssh access. See section 3 |  |
| Licensee | MarkLogic Licensee |  |
| LiceneKey | MarkLogic License Key |  |
| NodesPerZone | Nodes per AZ zone | 1 for instruction |
| PrivateSubnet1 | Internal Subnet for AZ #1 | rInternalSubnetAZ1 |
| PrivateSubnet2 | Internal Subnet for AZ #2 | rInternalSubnetAZ2 |
| PrivateSubnet3 | Internal Subnet for AZ #3 | rInternalSubnetAZ3 |
| SNSEmail | Email address to send diagnostic setup messages to. | Your email address or none if not required. |
| VolumeSize | Size of EBS volume in Gb | 10 for instruction |
| VolumeType | EBS volume type | gp2 |
| Zone1 | AZ corresponding to PrivateSubnet1[[11]](#footnote-11) | us-east-1a |
| Zone2 | AZ corresponding to PrivateSubnet2 | us-east-1b |
| Zone3 | AZ corresponding to PrivateSubnet3 | us-east-1d |

The SNSEmail parameter, if set, will be assigned to an SNS Topic[[12]](#footnote-12) resulting in diagnostic cluster setup messages reaching the specified address. There will be a large number of these sent, but it is useful to enable this when setting up for the first time, or in case of failure to setup correctly. Leave as ‘none’ if you do not wish to receive these messages.

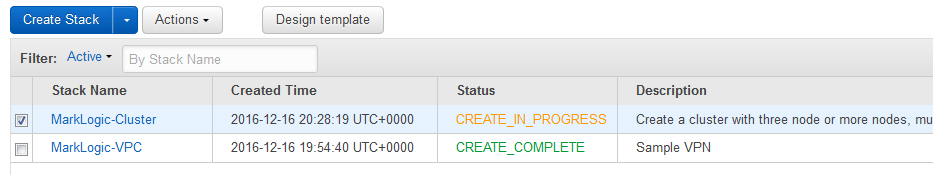
A populated template is shown below.



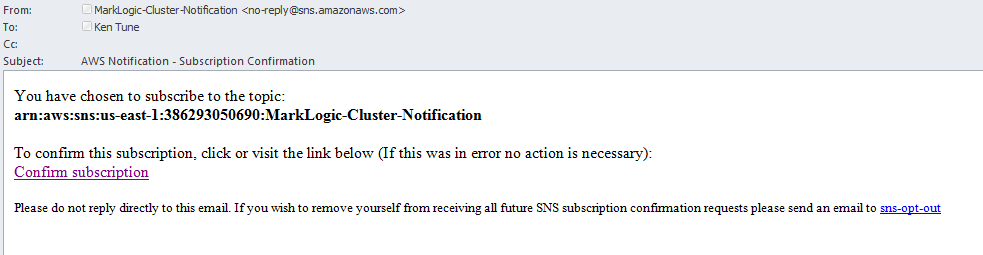
After populating the template, click ‘Next’ twice. At the bottom of the screen you will see



as the template creates an IAM role that governs permissions of the cluster nodes and this requires specific approval. Check the ‘I acknowledge’ box and click ‘Create’. You will see

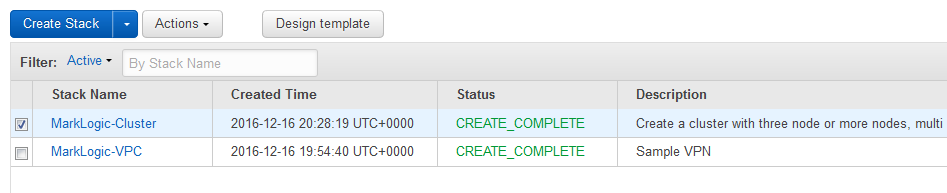


If you populated the SNSEmail field, shortly after, the specified address will receive a mail similar to



Click ‘Confirm Subscription’. If you do not, you will not receive diagnostic messages.

After around five minutes your template should complete and your Cloud Formation view should resemble



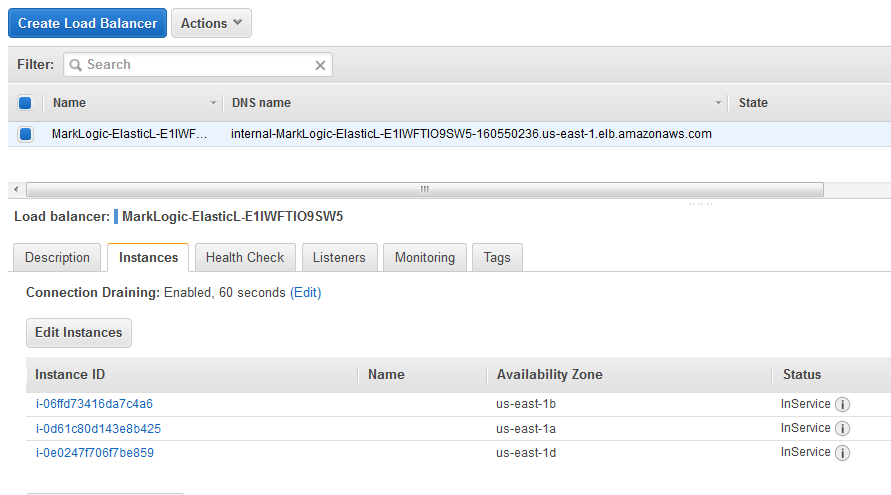
After the template completes setup of the cluster starts. The included Auto Scaling Groups[[13]](#footnote-13) will launch instances to match the ’Hosts per Zone’ parameter in the template and a number of operations will take place on each node in order to ready the cluster. If you provided an email address for the SNSEmail field, you will start to see diagnostic messages concerning the automated setup.

It will take 5 – 10 minutes from the point at which the cloud formation template completes before a fully available cluster is operational, or 10-15 minutes from the point at which the cloud formation stack creation was started.

# Verifying your Cluster

To check setup has completed successfully, we verify two things.

## All nodes in cluster available and connected to ELB[[14]](#footnote-14)

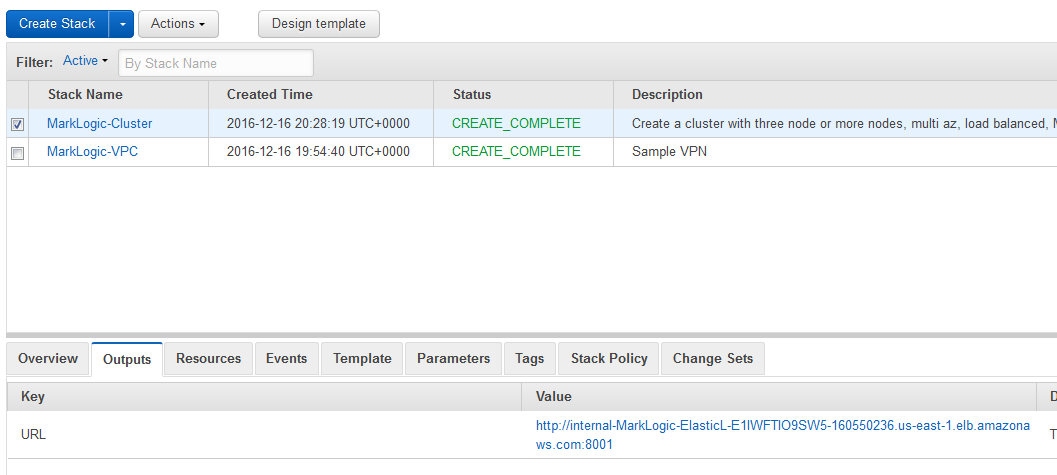
The first test of whether the cluster has set up correctly is whether the full cluster is available via the included Elastic Load Balancer. Go to the EC2 console - <https://console.aws.amazon.com/ec2> and select ‘Load Balancers’ from the left hand list. Select your load balancer followed by the ‘Instances’ tab. All three of our instances should be visible with status ‘In Service’, as below. 

If any are out of service, or absent then cluster setup has not completed.

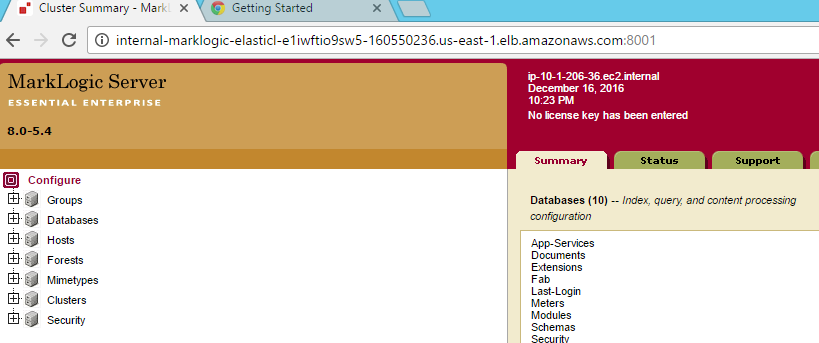
## Cluster is reachable via the ELB from the Windows Jump Box

The cluster should be reachable via the ELB from the Windows Jump box. If this is the case, then it will be reachable by any host in the rWebSecurityGroup, which as per section 6 would typically be the security group used for the middle tier.

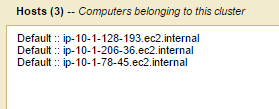
To verify this, first identify the ELB DNS name by going back to the Cloud Formation screen - <https://console.aws.amazon.com/cloudformation>, select the ‘MarkLogic Cluster’ stack and select the ‘Outputs’ tag.



Copy and paste your ELB URL somewhere convenient. Now log into the Windows jump box as per section 5.2, open Chrome and paste your ELB URL into the address bar. You should see the admin console as below. Note that you’re accessing it via your ELB URL.

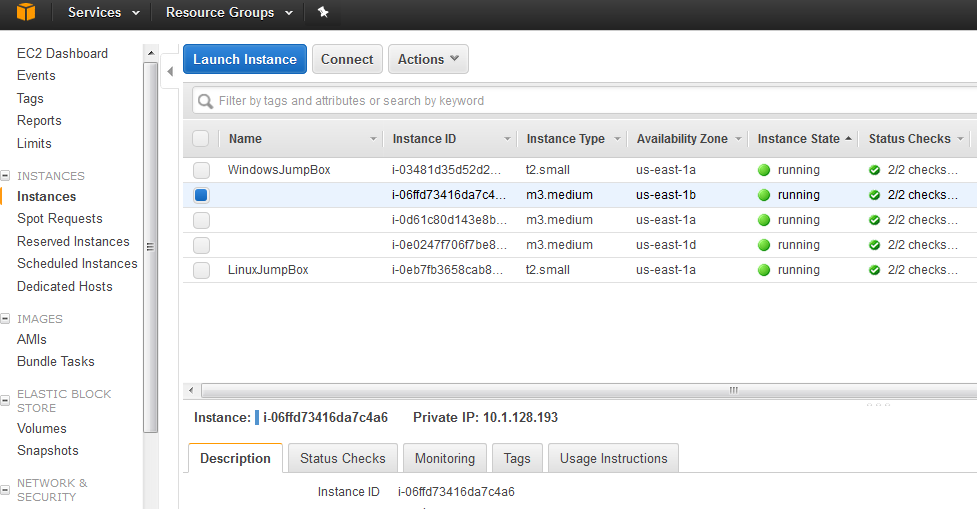


You should see your three connected hosts bottom right of this panel.



## Accessing hosts via ssh via Linux Jump Box

It’s also useful to verify that you can access your MarkLogic hosts at the command line. First obtain the IP of the hosts by visiting the EC2 console, <https://console.aws.amazon.com/ec2>, and clicking ‘Instances’. The IP addresses of the hosts may be found by selecting the hosts as shown. The first one below has IP 10.1.128.193.



To access this host, log into your Linux jump box as per section 5.1. Then ssh into the internal instance e.g. ssh 10.1.128.193. You should log in successfully, and also be able to do this for the other two boxes.

# Troubleshooting

VPCs can be difficult to troubleshoot precisely because they hide their contents from the outside world. Any difficulties encountered may well be associated with restrictions due to security groups, access control lists, or lack of public IPs. Check the implications of these carefully if resources are inaccessible.

If your MarkLogic cluster does not start successfully, the first course of action is to log into the MarkLogic hosts, if you can, and look at /var/log/messages[[15]](#footnote-15) for clues. Also look at the ‘Events’ tab associated with the Cloud Formation Stacks in the case of errors. The SNS messages are useful – look for ‘fail’. You will see that they are equivalent to the messages found in /var/log/messages.

# Conclusion

If you’ve got this far, then you’ve successfully built your managed MarkLogic Cluster inside a Virtual Private Cloud.

Note that there is more than one way to do this, and your own requirements may vary. Furthermore, for production use, you should review the detail thoroughly, especially the security aspects, to ensure configuration meets your requirements. Where applicable you should make changes – these examples are intended to help you get started rather than being rigid solutions.

Nevertheless, it is hoped that this document and the associated templates have been instructive.

1. Amazon Web Services [↑](#footnote-ref-1)
2. Virtual Private Cloud [↑](#footnote-ref-2)
3. <https://s3.amazonaws.com/marklogic-releases/8.0-5.4/ThreePlusCluster-BYOL.template>

   currently,or see <http://developer.marklogic.com/products/aws> if this URL changes [↑](#footnote-ref-3)
4. <http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html> [↑](#footnote-ref-4)
5. See <http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-using-cli.html> [↑](#footnote-ref-5)
6. A jump box is an intermediate host allowing indirect access to private resources. Also referred to as a bastion host. [↑](#footnote-ref-6)
7. From console.aws.amazon.com select Services, then click EC2 [↑](#footnote-ref-7)
8. As per section 5.1 [↑](#footnote-ref-8)
9. Other browsers are available. [↑](#footnote-ref-9)
10. Currently ~$0.05 per hour [↑](#footnote-ref-10)
11. Note it doesn’t seem to be possible to infer this from PrivateSubnet1 – at least I can’t make it happen. Fn::GetAtt operates on resources only, not parameters. [↑](#footnote-ref-11)
12. See <https://aws.amazon.com/sns/> [↑](#footnote-ref-12)
13. <https://aws.amazon.com/autoscaling/> [↑](#footnote-ref-13)
14. Elastic Load Balancer [↑](#footnote-ref-14)
15. Sudo required [↑](#footnote-ref-15)