Building a Modular and Scalable Virtual Network Architecture with Amazon VPC

Quick Start Reference Deployment

Santiago Cardenas
Solutions Architect, AWS Quick Start Reference Team

July 2016
Last updated: July 2017 (see revisions)

This guide is also available in HTML format at https://docs.aws.amazon.com/quickstart/latest/vpc/.

Contents

About This Guide

This Quick Start reference deployment guide discusses architectural considerations and configuration steps for deploying a modular Amazon Virtual Private Cloud (Amazon VPC) environment on the Amazon Web Services (AWS) Cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment.

The guide is for IT infrastructure architects, DevOps engineers, and administrators who would like to build a flexible, modular AWS networking infrastructure as a baseline for their deployments.

Quick Starts are automated reference deployments for AWS Cloud infrastructure components and key enterprise workloads on the AWS Cloud. Each Quick Start launches, configures, and runs AWS compute, network, storage, and other services, using AWS best practices for security and availability.

Overview

This Quick Start provides a networking foundation for AWS Cloud infrastructures. It deploys an Amazon Virtual Private Cloud (Amazon VPC) according to AWS best practices and guidelines. Amazon VPC is the networking layer for Amazon Elastic Compute Cloud (Amazon EC2) and provides a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network. For a discussion of best design practices for Amazon VPC environments, see the documentation and articles listed in the Additional Resources section.

The Amazon VPC architecture includes public and private subnets. The first set of private subnets share the default network access control list (ACL) from the Amazon VPC, and a second, optional set of private subnets includes dedicated custom network ACLs per subnet. The Quick Start divides the Amazon VPC address space in a predictable manner across multiple Availability Zones, and deploys either NAT instances or NAT gateways for outbound Internet access, depending on the AWS Region you deploy the Quick Start in.

You can use this Quick Start as a building block for your own deployments. You can scale it up or down by adding or removing subnets and Availability Zones according to your needs, and add other infrastructure components and software layers to complete your AWS environment.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, and other considerations discussed in this guide.

- If you have an AWS account and you're already familiar with AWS services, you can launch the Quick Start to build the architecture shown in Figure 1 in your AWS account. The deployment takes approximately 5 minutes. If you're new to AWS, please follow the step-by-step instructions provided later in this guide.

- If you want to take a look under the covers, you can view the AWS CloudFormation template that automates the deployment. The template includes default settings that you can customize by following the instructions in this guide.

Launch Quick Start | View template

Cost

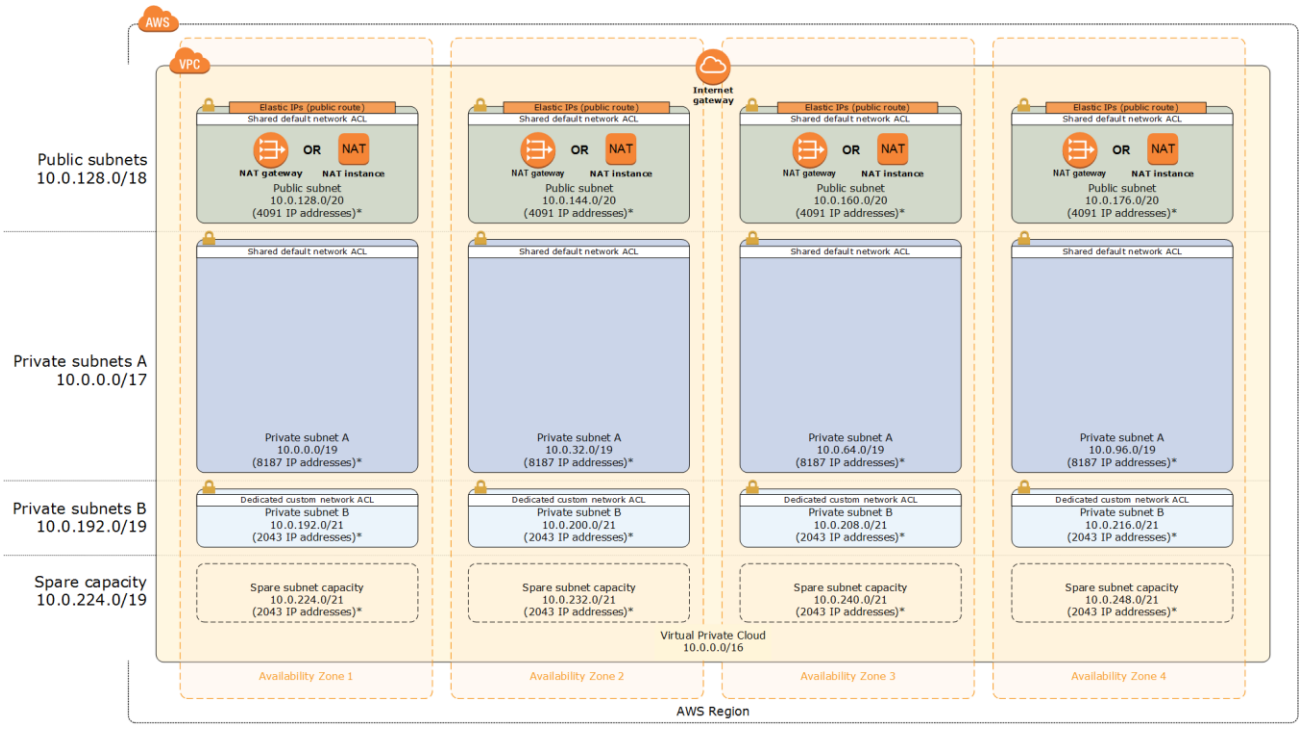
You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. Some of the settings, such as the instance type, number of Availability Zones, use of NAT gateways or NAT instances (depending on the AWS Region you choose), and amount of data that flows through the NAT device, will determine the cost of deployment. To get an estimate, you can use the AWS Simple Monthly Calculator.

For pricing details, see the Amazon EC2 pricing and Amazon VPC pricing pages.

Architecture

Deploying this Quick Start with the **default parameters** builds the following virtual networking environment in the AWS Cloud.



* Note that the IP addresses exclude five addresses from each subnet that are reserved and unavailable for use.

Figure 1: Modular Amazon VPC architecture on AWS (full-screen view)

The AWS CloudFormation template sets up the virtual network and creates networking resources.

The template creates a Multi-AZ, multi-subnet VPC infrastructure with managed NAT gateways in the public subnet for each Availability Zone. You can also create additional private subnets with dedicated custom network access control lists (ACLs). If you deploy the Quick Start in a region that doesn't support NAT gateways, NAT instances are deployed instead. Default subnet sizes are based on a typical deployment but can be reconfigured, as discussed in the Subnet Sizing section.

The Quick Start also includes VPC endpoints, which provide a secure, reliable connection to Amazon S3 without requiring an Internet gateway, a NAT device, or a virtual private gateway. With these endpoints, you can access S3 resources from within the VPC created by the Quick Start. These endpoints are valid only for the AWS Region in which you launch the Quick Start.

The Quick Start uses the default endpoint policy, which gives any user or service within the VPC full access to Amazon S3 resources. This policy supplements any IAM user policies or S3 bucket policies that you may have in place.

The Quick Start also enables Domain Name System (DNS) resolution in the VPC. For more information about VPC endpoints, see the AWS documentation.

AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the Getting Started section of the AWS documentation.)

- Amazon VPC – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of an IP address range, creation of subnets, and configuration of route tables and network gateways.
- Amazon EC2 – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- Amazon EBS – Amazon Elastic Block Store (Amazon EBS) provides persistent block-level storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect

you from component failure, offering high availability and durability. Amazon EBS volumes provide consistent and low-latency performance to run your workloads.

- NAT Gateway – NAT gateways are network address translation (NAT) devices, which provide outbound Internet access to instances in a private subnets, but prevent the Internet from accessing those instances. NAT gateways provide better availability and bandwidth than NAT instances. The NAT Gateway service is a managed service that takes care of administering NAT gateways for you. NAT gateways aren't supported in all AWS Regions. This Quick Start deploys NAT instances in regions where NAT gateways aren't available.

Best Practices

The architecture built by this Quick Start supports AWS best practices for high availability and security. The Quick Start provides:

- Up to four Availability Zones for high availability and disaster recovery. (AWS recommends maximizing your use of Availability Zones to isolate a data center outage.) Availability Zones are geographically distributed within a region and spaced for best insulation and stability in the event of a natural disaster.
- Separate subnets for unique routing requirements. AWS recommends using public subnets for external-facing resources and private subnets for internal resources. For each Availability Zone, this Quick Start provisions one public subnet and one private subnet by default. (If you need public subnets only, you can disable the creation of the private subnets.) For subnet sizing strategies, see the next section.
- Additional layer of security. AWS recommends using network ACLs as firewalls to control inbound and outbound traffic at the subnet level. This Quick Start provides an option to create a network ACL protected subnet in each Availability Zone. These network ACLs provide individual controls that you can customize as a second layer of defense.

  We recommend that you use network ACLs sparingly for the following reasons: they can be complex to manage, they are stateless, every IP address must be explicitly opened in each (inbound/outbound) direction, and they affect a complete subnet. We recommend that you use security groups more often than network ACLs, and create and apply these based on a schema that works for your organization. Some examples are server roles and application roles. For more information about security groups and network ACLs, see the Security section later in this guide.
- Independent route tables configured for every private subnet to control the flow of traffic within and outside the Amazon VPC. The public subnets share a single routing

table, because they all use the same Internet gateway as the sole route to communicate with the Internet.

- Highly available NAT gateways, where supported, instead of NAT instances. NAT gateways offer major advantages in terms of deployment, availability, and maintenance. For more information see the comparison provided in the Amazon VPC documentation.
- Spare capacity for additional subnets, to support your environment as it grows or changes over time.

For additional information about these best practices, see the following documentation:

- AWS Single VPC Design from the AWS Answers website
- Your VPC and Subnets in the Amazon VPC documentation
- Practical VPC Design in the AWS Startups blog
- Network ACLs in the Amazon VPC documentation

Subnet Sizing

In this Quick Start, the sizing of CIDR blocks used in the subnets is based on a typical deployment, where private subnets would have roughly double the number of instances found in public subnets. However, during deployment, you can use the CIDR block parameters to resize the CIDR scopes to meet your architectural needs.

In the default subnet allocation, the VPC is divided into subnet types and then further segmented per Availability Zone, as illustrated in Figure 1. The Quick Start provides the following default CIDR block sizes to maximize capacity:

VPC		**10.0.0.0/16**
Private subnets A	**10.0.0.0/17**	
	Availability Zone 1	10.0.0.0/19
	Availability Zone 2	10.0.32.0/19
	Availability Zone 3	10.0.64.0/19
	Availability Zone 4	10.0.96.0/19
Public subnets	**10.0.128.0/18**	
	Availability Zone 1	10.0.128.0/20
	Availability Zone 2	10.0.144.0/20
	Availability Zone 3	10.0.160.0/20
	Availability Zone 4	10.0.176.0/20

VPC		10.0.0.0/16
Private subnets B with dedicated custom network ACL	**10.0.192.0/19**	
	Availability Zone 1	10.0.192.0/21
	Availability Zone 2	10.0.200.0/21
	Availability Zone 3	10.0.208.0/21
	Availability Zone 4	10.0.216.0/21
Spare subnet capacity	**10.0.224.0/19**	
	Availability Zone 1	10.0.224.0/21
	Availability Zone 2	10.0.232.0/21
	Availability Zone 3	10.0.240.0/21
	Availability Zone 4	10.0.248.0/21

Alternatively, there may be situations where you would want to separate the CIDR scopes by dividing the VPC into Availability Zones and then into subnet types. The recommended CIDR blocks to maximize capacity for this scenario are as follows:

VPC		10.0.0.0/16
Availability Zone 1	**10.0.0.0/18**	
	Private subnet A	10.0.0.0/19
	Public subnet	10.0.32.0/20
	Private subnet B	10.0.48.0/21
	Spare subnet capacity	10.0.56.0/21
Availibility Zone 2	**10.0.64.0/18**	
	Private subnet A	10.0.64.0/19
	Public subnet	10.0.96.0/20
	Private subnet B	10.0.112.0/21
	Spare subnet capacity	10.0.120.0/21
Availability Zone 3	**10.0.128.0/18**	
	Private subnet A	10.0.128.0/19
	Public subnet	10.0.160.0/20
	Private subnet B	10.0.176.0/21
	Spare subnet capacity	10.0.184.0/21

VPC	10.0.0.0/16	
Availability Zone 4	**10.0.192.0/18**	
	Private subnet A	10.0.192.0/19
	Public subnet	10.0.224.0/20
	Private subnet B	10.0.240.0/21
	Spare subnet capacity	10.0.248.0/21

To customize the CIDR ranges for this scenario or to implement your own segmentation strategy, you can configure the Quick Start parameters described in step 2. For more information about VPC and subnet sizing, see the AWS documentation.

Deployment Steps

Follow the step-by-step instructions in this section to build the virtual network environment illustrated in Figure 1 in your AWS account. The AWS CloudFormation template provided with this Quick Start bootstraps the AWS networking infrastructure on the AWS Cloud from scratch.

Step 1. Prepare an AWS Account

1. If you don't already have an AWS account, create one at http://aws.amazon.com by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy the Quick Start on AWS.

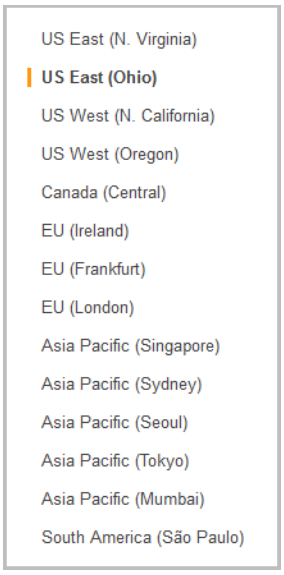


Figure 2: Choosing an AWS Region

Tip Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

Also, note that your choice of region will determine whether the Quick Start deploys NAT gateways or NAT instances for network connections. For a list of regions that support NAT gateways, see Amazon VPC pricing.

3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

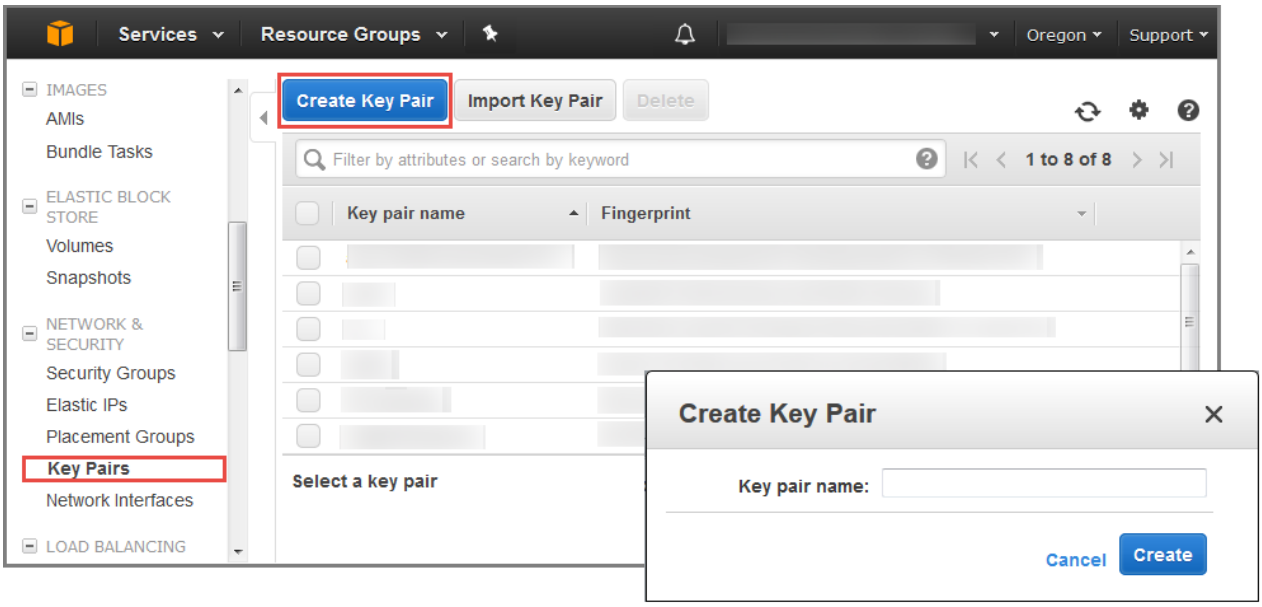


Figure 3: Creating a key pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. With Windows instances, we use the key pair to obtain the administrator password via the Amazon EC2 console and then log in using Remote Desktop Protocol (RDP) as explained in the step-by-step instructions in the *Amazon Elastic Compute Cloud User Guide*. On Linux, we use the key pair to authenticate SSH login.

Step 2. Launch the Stack

1. Launch the Quick Start into your AWS account.

   Launch

   The template is launched in the US West (Oregon) region by default. You can change the region by using the region selector in the navigation bar.

   This stack takes approximately 5 minutes to create.

   Note You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. See the pricing pages for each AWS service you will be using in this Quick Start for full details.

2. On the **Select Template** page, keep the default setting for the Amazon S3 template URL, and then choose **Next**.

3. On the **Specify Details** page, review the following parameters for the template, provide values for parameters that require your input, and customize the default settings as necessary. For example, you can change the network configuration parameters if you want to reconfigure the subnet segmentation used for the VPC, as discussed earlier in the Subnet Sizing section.

Availability Zone Configuration:

Parameter label (name)	Default	Description
Availability Zones (AvailabilityZones)	*Requires input*	The specific Availability Zones you want to use for resource distribution. This field displays the available zones within your selected region. You can choose 2, 3, or 4 Availability Zones from this list. The logical order of your selections is preserved in your deployment. After you make your selections, make sure that the value of the **Number of Availability Zones** parameter matches the number of selections.
Number of Availability Zones (NumberOfAZs)	*Requires input*	The number of Availability Zones you want to use in your deployment, to ensure high availability of resources. You can specify 2, 3, or 4 Availability Zones. This count must match the number of selections you make from the **Availability Zones** parameter; otherwise, your deployment will fail with an AWS CloudFormation template validation error. (Note that some regions provide only 2 or 3 Availability Zones.)

Network Configuration:

Parameter label (name)	Default	Description
VPC CIDR (VPCCIDR)	10.0.0.0/16	CIDR block for the Amazon VPC.
Public subnet 1 CIDR (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for public (DMZ) subnet 1 located in Availability Zone 1.
Public subnet 2 CIDR (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for public (DMZ) subnet 2 located in Availability Zone 2.
Public subnet 3 CIDR (PublicSubnet3CIDR)	10.0.160.0/20	CIDR block for public (DMZ) subnet 3 located in Availability Zone 3.
Public subnet 4 CIDR (PublicSubnet4CIDR)	10.0.176.0/20	CIDR block for public (DMZ) subnet 4 located in Availability Zone 4.
Create private subnets (CreatePrivateSubnets)	true	Set to **false** to create only public subnets in the VPC. If **true**, the CIDR blocks for those subnets will be determined by the following four parameters. If **false**, the CIDR parameters for all private subnets will be ignored.

Parameter label (name)	Default	Description
Private subnet 1A CIDR (PrivateSubnet1ACIDR)	10.0.0.0/19	CIDR block for private subnet 1A located in Availability Zone 1.
Private subnet 2A CIDR (PrivateSubnet2ACIDR)	10.0.32.0/19	CIDR block for private subnet 2A located in Availability Zone 2.
Private subnet 3A CIDR (PrivateSubnet3ACIDR)	10.0.64.0/19	CIDR block for private subnet 3A located in Availability Zone 3.
Private subnet 4A CIDR (PrivateSubnet4ACIDR)	10.0.96.0/19	CIDR block for private subnet 4A located in Availability Zone 4.
Create additional private subnets with dedicated network ACLs (CreateAdditionalPrivateSubnets)	false	Set to **true** to create a private subnet with dedicated network ACL in each Availability Zone for additional security. If **true**, the IP address ranges for the CIDR block will be determined by the following four parameters. If **false** (default), the CIDR parameters for those subnets will be ignored. See the Security section to read about using network ACLs vs. security groups.
Private subnet 1B with dedicated network ACL CIDR (PrivateSubnet1BCIDR)	10.0.192.0/21	CIDR block for private subnet 1B with dedicated network ACL, located in Availability Zone 1.
Private subnet 2B with dedicated network ACL CIDR (PrivateSubnet2BCIDR)	10.0.200.0/21	CIDR block for private subnet 2B with dedicated network ACL, located in Availability Zone 2.
Private subnet 3B with dedicated network ACL CIDR (PrivateSubnet3BCIDR)	10.0.208.0/21	CIDR block for private subnet 3B with dedicated network ACL, located in Availability Zone 3.
Private subnet 4B with dedicated network ACL CIDR (PrivateSubnet4BCIDR)	10.0.216.0/21	CIDR block for private subnet 4B with dedicated network ACL, located in Availability Zone 4.
VPC Tenancy (VPCTenancy)	default	The tenancy attribute for the instances launched into the VPC. By default, all instances in the VPC run as shared-tenancy instances. Set this parameter to **dedicated** to run them as single-tenancy instances instead. For more information, see Dedicated Instances in the *Amazon EC2 User Guide*.

Amazon EC2 Configuration:

Parameter label (name)	Default	Description
Key pair name (KeyPairName)	*Requires input*	Public/private key pair, which allows you to connect securely to your instance after it launches. When you

Parameter label (name)	Default	Description
		created an AWS account, this is the key pair you created in your preferred region.
NAT instance type (NATInstanceType)	t2.small	EC2 instance type for NAT instances. This value is used only when the Quick Start deploys NAT instances, when the AWS Region you selected doesn't support NAT gateways.

When you finish reviewing and customizing the parameters, choose **Next**.

> **Note** You can also download the template and edit it to create your own parameters based on your specific deployment scenario.

4. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you're done, choose **Next**.
5. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
6. Choose **Create** to deploy the stack.
7. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the stack is ready.

Step 3. Add AWS Services or Other Applications

After you use this Quick Start to build your VPC environment, you can deploy additional Quick Starts or deploy your own applications on top of this AWS infrastructure. If you decide to extend your AWS environment with additional Quick Starts for trial or production use, we recommend that you choose the option to deploy the Quick Start into an existing VPC, where that option is available.

If you decide to deploy additional private subnets with dedicated network ACLs, make sure you review the configuration and adjust it accordingly. By default, the custom ACLs are configured to allow all inbound and outbound traffic to flow in order to facilitate the deployment of additional infrastructure. For more information, see Network ACLs and Recommended Network ACL Rules for Your VPC in the Amazon VPC documentation.

Troubleshooting

When you deploy this Quick Start, if you encounter a **CREATE_FAILED** error instead of the **CREATE_COMPLETE** status code, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state will be retained and the instance will be left running, so you can troubleshoot the issue. (You'll want to look at the log files in `%ProgramFiles%\Amazon\EC2ConfigService` and in the `C:\cfn\log` folder.)

> **Important** When you set **Rollback on failure** to **No**, you'll continue to incur AWS charges for this stack. Please make sure to delete the stack when you've finished troubleshooting.

The following table lists specific **CREATE_FAILED** error messages you might encounter.

Error message	Possible cause	What to do
API: ec2: RunInstances Not authorized for images: *ami-ID*	The template is referencing an AMI that has expired.	We refresh AMIs on a regular basis, but our schedule isn't always synchronized with AWS AMI updates. If you get this error message, notify us, and we'll update the template with the new AMI ID. If you'd like to fix the template yourself, you can download it and update the `Mappings` section with the latest AMI ID for your region.
We currently do not have sufficient t2.small capacity in the AZ you requested	The NAT instance requires a larger or different instance type.	Switch to an instance type that supports higher capacity. If a higher-capacity instance type isn't available, try a different Availability Zone or region. Or you can complete the request form in the AWS Support Center to increase the Amazon EC2 limit for the instance type or region. Limit increases are tied to the region they were requested for.
Instance *ID* did not stabilize	You have exceeded your IOPS for the region.	Request a limit increase by completing the request form in the AWS Support Center.

If you encounter a template validation error during deployment, check for a mismatch in the values of the **Availability Zones** and **Number of Availability Zones** parameters. If you select more Availability Zones than you request, the AWS CloudFormation template won't validate. Correct the parameters so that they're in sync, and redeploy the Quick Start.

For additional information, see Troubleshooting AWS CloudFormation on the AWS website. If the problem you encounter isn't covered on that page or in the table, please visit the AWS Support Center. If you're filing a support ticket, please attach the install.log file from the master instance (this is the log file that is located in the `/root/install` folder) to the ticket.

Security

Public and Private Subnets

This Quick Start provisions one public and one private subnet in each Availability Zone by default. You can also choose to add additional private subnets with dedicated network ACLs.

A public subnet is directly routable to the Internet via a route in the route table that points to the Internet gateway. This type of subnet allows the use of Elastic IPs and public IPs, and (if the security group and network ACLs permit) a public subnet is reachable from the Internet. A public subnet is useful as a DMZ infrastructure for web servers and for Internet-facing Elastic Load Balancing (ELB) load balancers.

Private subnets can indirectly route to the Internet via a NAT instance or NAT gateway. These NAT devices reside in a public subnet in order to route directly to the Internet. Instances in a private subnet are not externally reachable from outside the Amazon VPC, regardless of whether they have a public or Elastic IP address attached. A private subnet is useful for application servers and databases.

Using Security Groups and Network ACLs

The following table (reprinted here from the AWS documentation for convenience) describes the differences between security groups and network ACLs:

Security group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules

Security group	Network ACL
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in numerical order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group)

The network ACLs in this Quick Start are configured as follows:

- All public and private subnets are associated with the same default network ACL, which is automatically created for all VPCs on AWS. This network ACL allows all inbound and outbound traffic. As you deploy instances and services, you should associate them with security groups and allow only the traffic and ports needed for your application.
- Each additional private subnet is associated with a custom network ACL (1:1 ratio). These network ACLs are initially configured to allow all inbound and outbound traffic to facilitate the deployment of additional instances and services. As with the other subnets, you should use security groups to secure the environment internally, and you can lock down the custom network ACLs during or after deployment as required by your application.

If the Quick Start deploys NAT instances instead of NAT gateways in the AWS Region you selected, it adds a single security group as a virtual firewall. This security group is required for NAT instances and any other instances in the private subnets to access the Internet. The security group is configured as follows:

Inbound:

Source	Protocol	Ports
VPC CIDR	All	All

Outbound:

Destination	Protocol	Ports
0.0.0.0/0	All	All

For additional details, see Security in Your VPC in the Amazon VPC documentation.

Additional Resources

AWS services

- AWS CloudFormation
  http://aws.amazon.com/documentation/cloudformation/
- Amazon EC2
  - User guide for Microsoft Windows:
    http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/
  - User guide for Linux:
    https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
- Amazon VPC
  http://aws.amazon.com/documentation/vpc/
  - Security groups
    https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html
  - Network ACLs
    https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html
  - NAT gateways
    http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html
- Best practices for implementing VPCs
  - AWS Single VPC Design
    http://do.awsstatic.com/aws-answers/AWS_Single_VPC_Design.pdf
  - Your VPC and Subnets
    http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html
  - Practical VPC Design
    https://medium.com/aws-activate-startup-blog/practical-vpc-design-8412e1a18dcc

Quick Start reference deployments

- AWS Quick Start home page
  https://aws.amazon.com/quickstart/

Send Us Feedback

You can visit our GitHub repository to download the templates and scripts for this Quick Start, to post your comments, and to share your customizations with others.

Document Revisions

Date	Change	In sections
July 2017	Added new **CreatePrivateSubnets** and **VPCTenancy** parameters.	Step 2 parameter table
August 2016	Added VPC endpoints for Amazon S3	Template changes Architecture section
July 2016	Instances launched in the public subnets will, by default, get a public IP address Added discussion of subnet segmentation strategies; changed CIDR range defaults	Template changes Subnet Sizing section
July 2016	Initial publication	—

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The software included with this paper is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at http://aws.amazon.com/apache2.0/ or in the "license" file accompanying this file. This code is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.