Running IxLoad in the Cloud

Release 8.50

Notices

Copyright Notice

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula or https://support.ixiacom.com/support-services/warranty-license-agreements. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Key-sight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contact Us

Ixia headquarters

26601 West Agoura Road
Calabasas, California 91302
+1 877 367 4942 – Toll-free North America
+1 818 871 1800 – Outside North America
+1.818.871.1805 – Fax
www.ixiacom.com/contact/info

Support

Global Support	+1 818 595 2599	support@ixiacom.com
APAC Support	+91 80 4939 6410	support-asiapac@ixiacom.com
EMEA Support	+40 21 301 5699	support-emea@ixiacom.com
Greater China Region	+400 898 0598	support-china@ixiacom.com
India Office	+91 80 4939 6410	support-india@ixiacom.com
Japan Head Office	+81 3 5326 1980	support-japan@ixiacom.com
Korea Office	+82 2 3461 0095	support-korea@ixiacom.com
Singapore Office	+656 494 8910	support-asiapac@ixiacom.com

This page intentionally left blank.

CONTENTS

Introduction

AWS's CloudFormation Templates (CFT) simplify the process of provisioning and management on AWS.

Templates are configure to create for the services or applications desired, and AWS CloudFormation uses those templates quickly and reliably provision of the services or applications (called “stacks”).

Ixia has created a CloudFormation template to enable deployment of IxLoad in AWS. The template deploys IxVM instances and DUT instances in a new VPC.

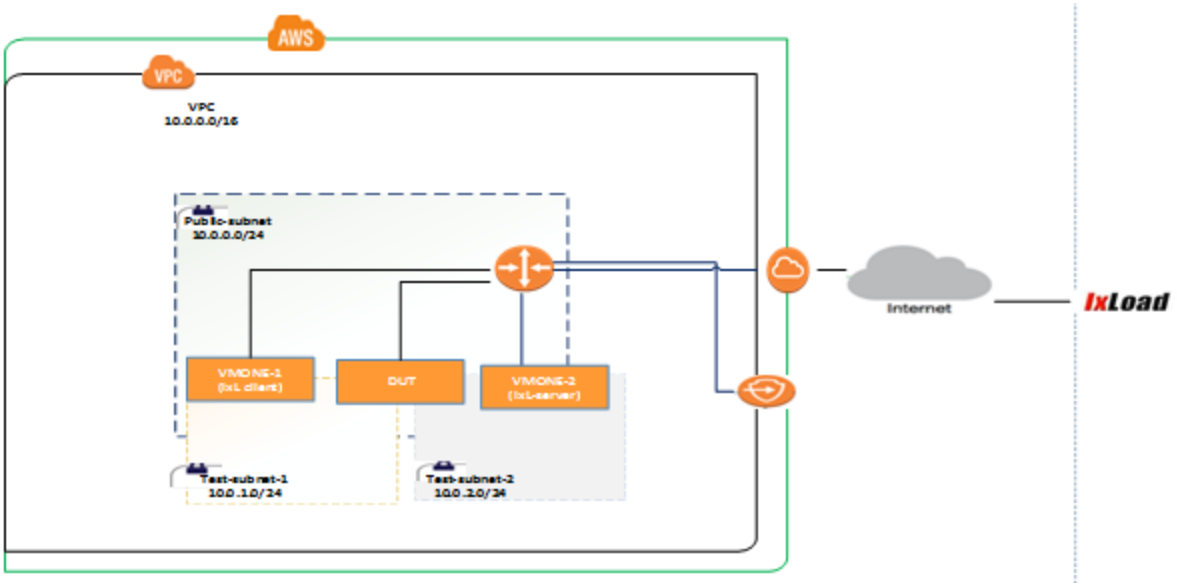
Every newly-deployed instance has one public network and 1 to 8 test network ports, each with a private IP address.

In addition to the instances, the IxLoad CloudFormation template automatically creates the following AWS entities:

- One VPC with same name as given for Stack name.
- One public subnet and one private subnets for each of the test interface specified during stack creation.
- Two new routing tables named with the stack name. One table is for public routing and the other for private routing. All private subnets are associated with the private routing table.
- An internet gateway that is assigned to the public routing table and public subnet
- Two security groups named with the stack name. One group is the for management interface and the other is for the test interfaces.
- Two IAM roles, EC2ReadonlyRole and LambdaExecutionRole ,named with stack name. These roles are assigned to newly created virtual test appliance instances.

The image below shows the IxLoad AWS topology.

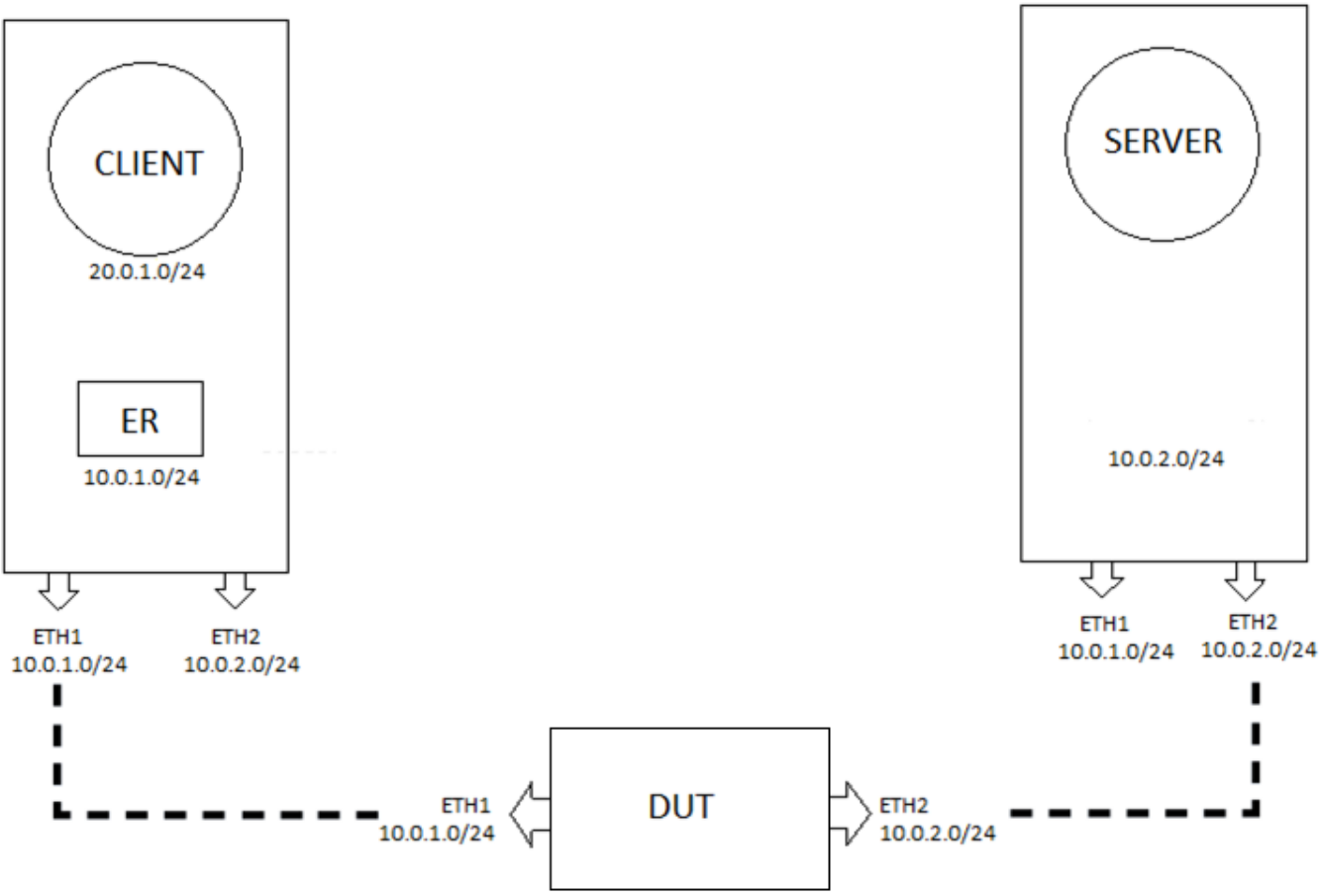
IxLoad AWS topology



IxLoad test topology

You can configure the client network with or without an Emulated Router (ER).

The image below shows an IxLoad client NetTraffic connected to a server NetTraffic through a DUT.



Deployment

This section describes how to use the Ixia CloudFormation template to deploy and configure IxVM virtual test appliance instances on AWS.

Before you begin

Before you begin, you must:

- have an AWS user ID that has IAM Role Creation and Lambda Execution privileges
- create a key pair before using CFT
- know the AMI IDs of the virtual test appliance image and the DUT image. These images should be visible in the AMI bucket of the required region. To find the virtual test appliance AMI ID, open the AWS community, filter for `ixia` and select Ixia Virtual Test Appliance <version> AMI ID.
- know the external IP (NAT) address of the network that you use to access AWS. You can get this address from your IT department, or you can get it by using the following procedure:
  1. In your AWS account, create a new security rule
  2. For Source, select My IP.

     AWS will display your network's external IP.

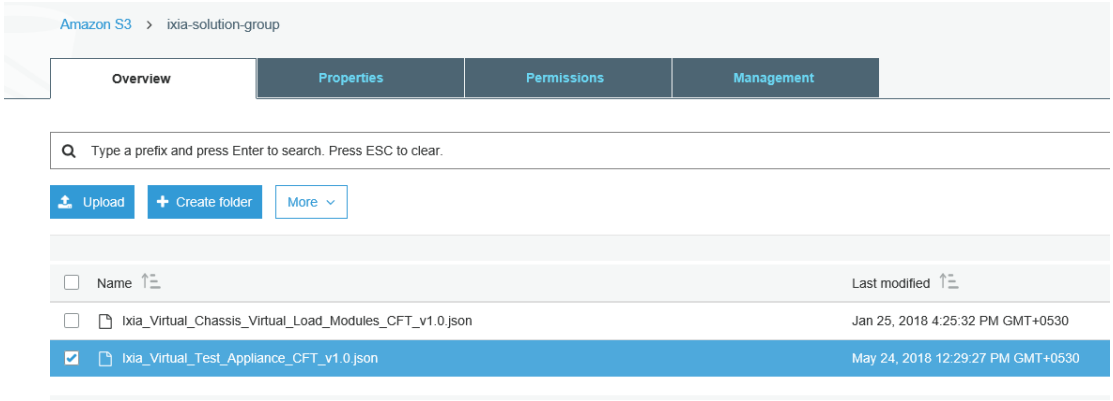
After deploying the template, you must configure the secondary IP addresses of each instance, either manually or by using a script.

Deployment

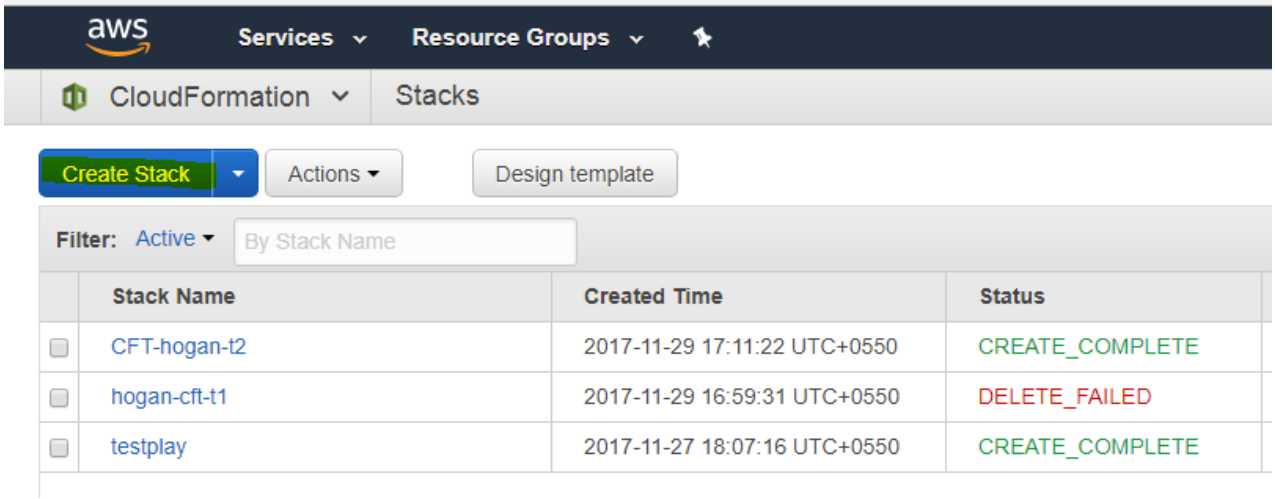
To deploy the template:

1. Login to AWS.
2. Select the json template file.

   You can either use the pre-configured Ixia S3 template json file or you can use your own.
   - If you want to use the Ixia json file, its url is: `https://s3.amazonaws.com/ixia-solution-group/Ixia_Virtual_Test_Appliance_CFT_v1.1.json`
   - If you want to use your own, upload it to your local S3 bucket and get the url for the json file.

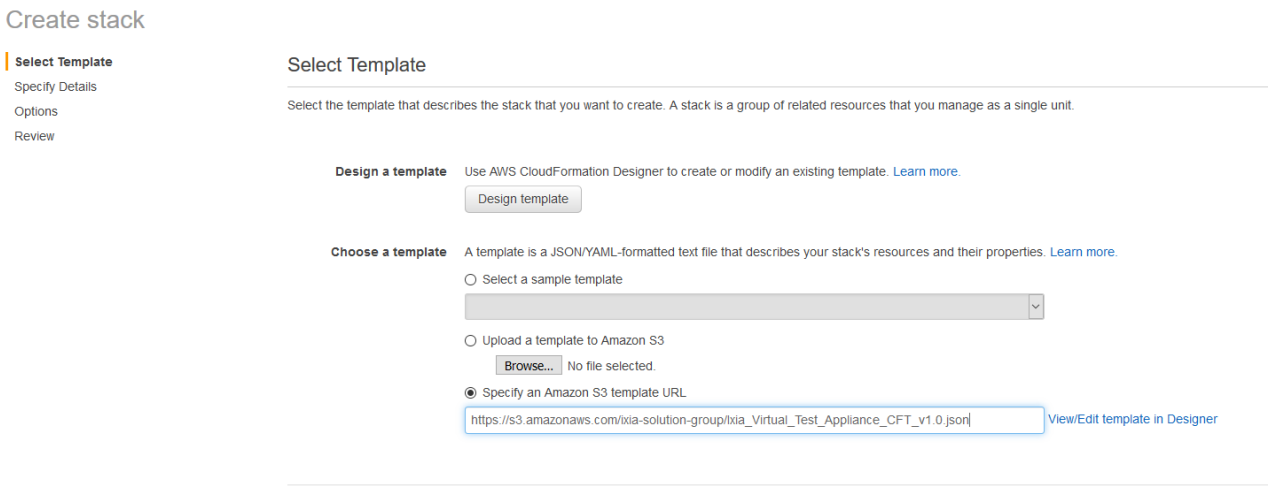


3. Click Services | CloudFormation, then click Create Stack.



The Create Stack page displays.

4. Under Choose a template, click Specify an Amazon S3 template URL, then enter the URL of the json file you want to use.



5. Click Next.

   The Specify Details page displays.
6. Enter the parameters for the VMOne instance.

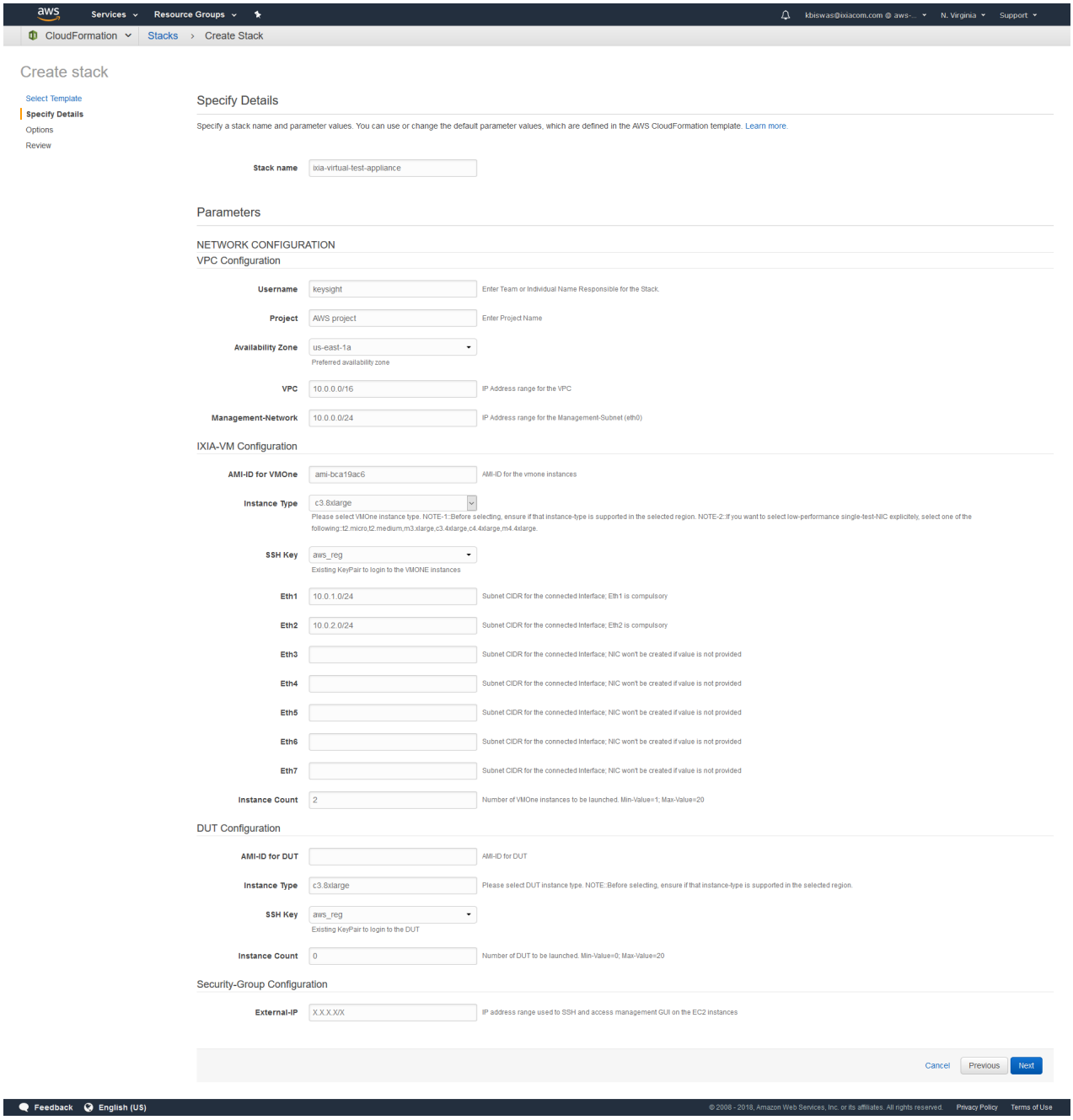
   Observe the following guidelines:

   DUT:

   - If you intend to deploy a DUT, configure 2 interfaces (Eth1 and Eth2) under the IXIA-VM Configuration section.
   - If you do not intend to deploy a DUT, specify 0 for "Instance count" under "DUT Configuration"

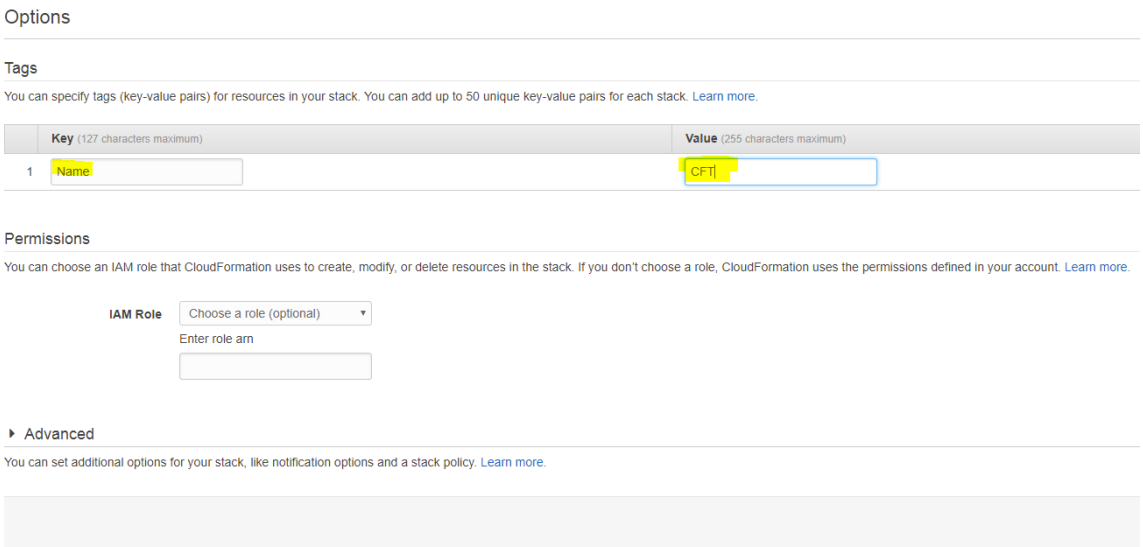
   Enhanced Networking:

   - If you want to use an instance that has SR-IOV (Enhanced Networking) enabled, select "C3.8xlage" in the Instance type field.
   - If you do not need an instance that has SR-IOV enabled, select "m3.xlarge" in the Instance type field.
   - For a list of other instance types that support SR-IOV, see "Enhanced Networking Types" in http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html.
   - Make sure you enter your network's external IP (NAT) that you use to access AWS.
7. After you have entered the VMOne parameters, click Next.

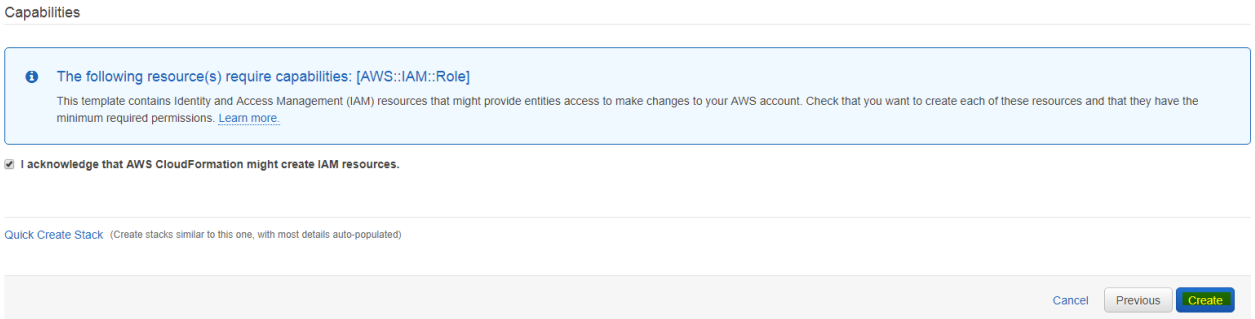


The Options page displays.

8. Enter a tag a (key-value pair) that you can use to identify the instance.

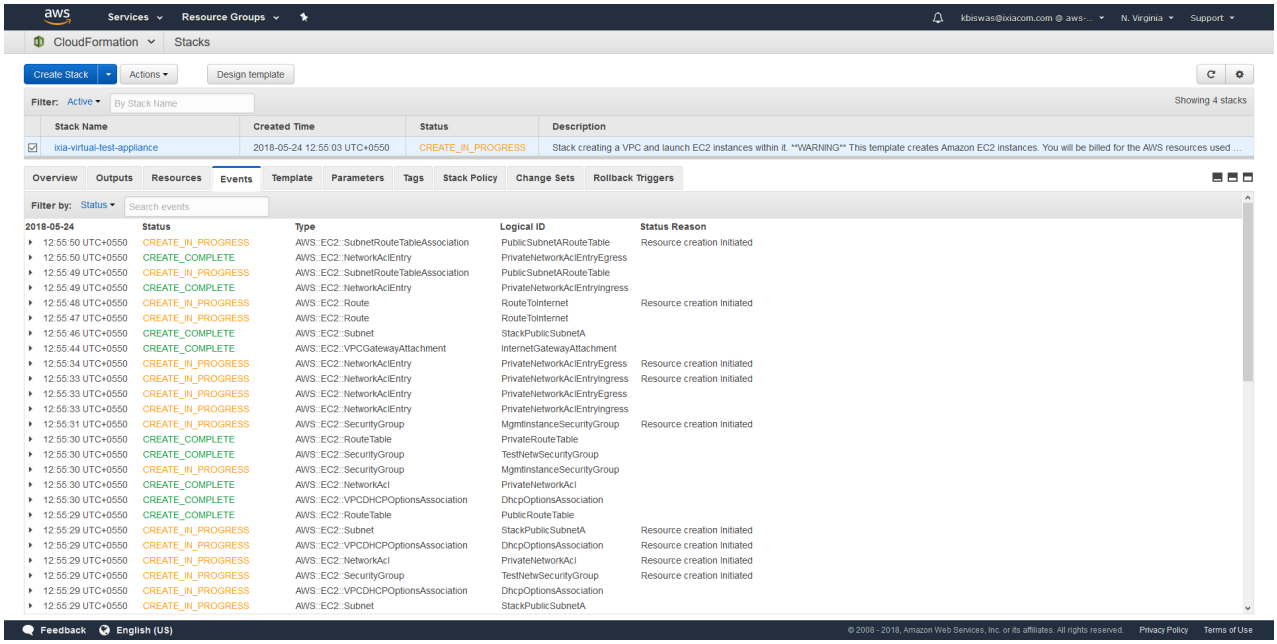


9. Under Capabilities, click the checkbox for "I acknowledge that AWS CloudFormation might create IAM resources. The template needs this permission to access and control custom resources through the lambda function.



10. Click Create.
    AWS begins deploying the stack.

    On the CloudFormation Stacks page, the stack deployment status will show `CREATE_IN_PROGRESS`.
    You can click the Events tab to view the status of various sub-tasks.
    When the stack has been deployed, the stack status displays as `CREATE_COMPLETE`.



11. After the stack is deployed, click the Outputs tab and make a note of:
    - the login information for each instance.
    - the `VMONE INSTANCE DETAILS`, which displays the IP address details in json format

    For example, in the deployment shown in the image below, there are two Virtual Test Appliances:
    `IXIA-VMONE-i-0b1fc16f1ef2c6852`

    and

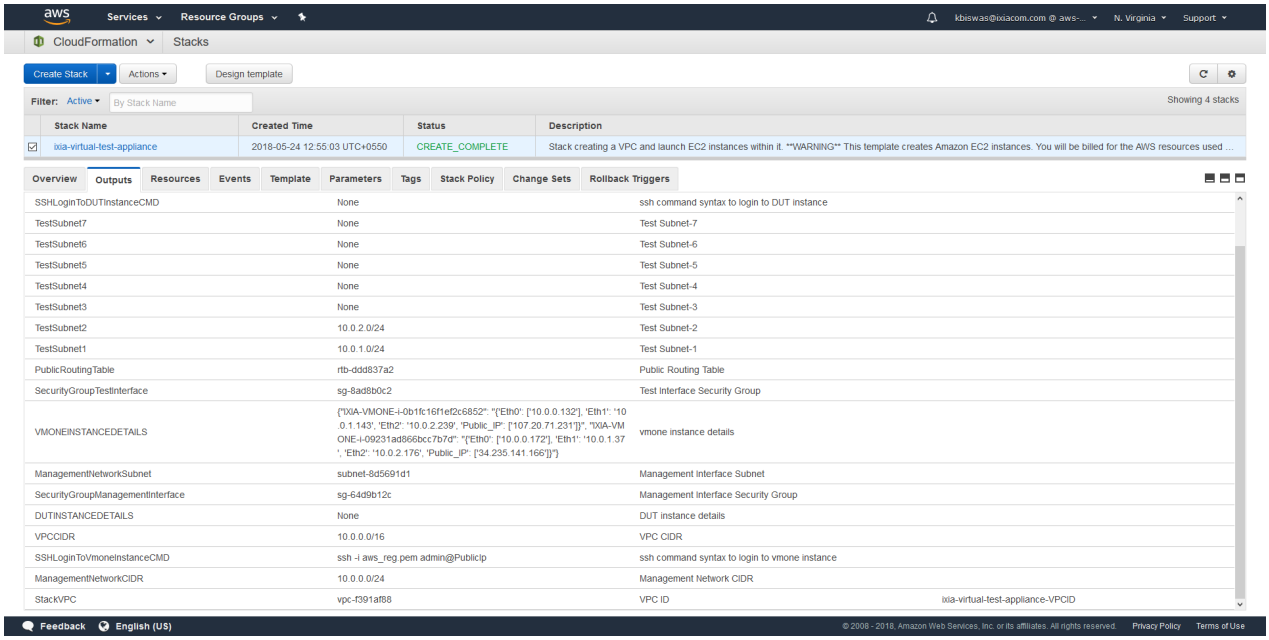
    `IXIA-VMONE-i-09231ad866bcc7b7d`

    Each Virtual Test Appliance has three interfaces: - one management interface (Eth0) and two test interfaces. There are two test interfaces because when the appliance was configured from the CFT, two test interfaces, Eth1 and Eth2, were specified. .

    Each Virtual Test Appliance displays its IP address list. For example "IXIA-VMONE-i-0b1fc16f1ef2c6852" has IP list "{'Eth0': ['10.0.0.132'], 'Eth1': '10.0.1.143', 'Eth2': '10.0.2.239', 'Public_IP': ['107.20.71.231']}".

    Eth0 interface has a private IP address of 10.0.0.132 , which is associated with public IP address 107.20.71.231. This virtual test appliance can be added to IxLoad using this public IP address.

    Eth1 interface has a private IP address of 10.0.1.143, and Eth2 has a private IP address of 10.0.2.239. These are the IP addresses that must be configured as test ports in IxLoad.



Security groups

The CloudFormation template automatically creates security groups and configures the ports allowed for the inbound and outbound rules for both the management and the test interfaces.

Management interfaces

For the management interfaces, the ports allowed are:

Inbound rule:

TCP Ports	22, 80, 443, 111, 2601, 998-999, 1000, 1080, 2345, 3222, 3601, 4501, 4502, 4601, 5285, 5286, 5236, 5237, 5480, 5488, 5489, 6001-6005, 6665, 6967, 6978, 8021, 8022, 8881, 8989, 8990, 9101, 9102, 9613-9676, 10115, 10116, 10119, 17662, 17668-17777, 18765, 21123, 21653
UDP Ports	67, 68, 123, 161, 162, 605, 1000, 6004 ,10116
ICMP	<none>

Outbound rule:

ICMP	<none>

Test interfaces

Inbound rule: All ports

Outbound rule: All ports

Note:
IxLoad uses the `boto3` python library to extract AWS-related information.

`boto3` uses public IP addresses that vary by region to communicate with AWS. These IPs are listed on the following page:

https://ip-ranges.amazonaws.com/ip-ranges.json

You must manually add the required IPs to virtual test appliance's management security group manually.

For more information, see:

http://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html

Configuring secondary IP addresses

By default, the CloudFormation template configures only one IP address per interface. If you need to use multiple IP addresses per interface, you must configure secondary IP addresses.

You can configure secondary IP addresses manually or automatically.

Manually

Refer to this page in the AWS documentation for the procedure on configuring secondary IP addresses: https://aws.amazon.com/premiumsupport/knowledge-center/secondary-private-ip-address/

Automatically

Create a python script to automatically configure secondary IP addresses.

1. Configure the AWS CLI. Refer to this page in the AWS documentation: https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-linux.html#awscli-install-linux-path
2. Install `boto3`:
   ```
   # pip install boto3 --user
   ```
3. After installing `boto3`, configure AWS CLI with the `AWS configure` command along with the AWS Access Key ID and AWS Secret Access Key.
   ```
   [ec2-user@ip-10-0-0-5 ~]$ aws configure
   AWS Access Key ID [*****************I2BQ]
   AWS Secret Access Key [*****************y
   ```
4. Create a python script with content listed below.
5. Execute the script with required parameters:

```
python createCustomPrivateIP.py [instance-id] [eth1/eth2/...] [start_ip] [stop_
ip] [Region Name]
```

Script

```
## USAGE  : python createCustomPrivateIP.py [instance-id] [eth1/eth2/...] [start_ip]
[stop_ip] [Region Name]
## EXAMPLE: python createCustomPrivateIP.py i-0cba4998da080d95e eth1 10.0.1.37
10.0.1.39 us-east-1

import sys
import boto3
from botocore.exceptions import ClientError

client = boto3.client('ec2', region_name= sys.argv[5])
ec2 = boto3.resource('ec2', region_name= sys.argv[5])

start_ip = sys.argv[3]     #start_ip = '172.31.15.171'
stop_ip  = sys.argv[4]     #stop_ip  = '172.31.15.174'

## Create IP-Address-Range
def undotIPv4 (dotted):
    return sum (int (octet) << ( (3 - i) << 3) for i, octet in enumerate
(dotted.split ('.') ) )
def dotIPv4 (addr):
    return '.'.join (str (addr >> off & 0xff) for off in (24, 16, 8, 0) )
def rangeIPv4 (start, stop):
    for addr in range (undotIPv4 (start), undotIPv4 (stop) ):
        yield dotIPv4 (addr)

if (sys.argv[2] == 'eth1'):
    device_index=1
elif (sys.argv[2] == 'eth2'):
    device_index=2
elif (sys.argv[2] == 'eth3'):
    device_index=3
elif (sys.argv[2] == 'eth4'):
    device_index=4
elif (sys.argv[2] == 'eth5'):
    device_index=5
elif (sys.argv[2] == 'eth6'):
    device_index=6
elif (sys.argv[2] == 'eth7'):
    device_index=7
else:
    device_index=8
```

```
try:
    vpc_instances = ec2.network_interfaces.filter(Filters=[{'Name':
'attachment.instance-id','Values': [sys.argv[1]]}, {'Name': 'attachment.device-
index','Values': [str(device_index)]}])
    for interfaces in vpc_instances:
        for ip_list in rangeIPv4 (start_ip, stop_ip):
            response1 = client.assign_private_ip_addresses
(NetworkInterfaceId=interfaces.id, PrivateIpAddresses=[ip_list],)
            print(response1)
except ClientError as e:
    print(e)
```

Configuring an IxLoad test in AWS

After using the CFT to deploy your AWS instances, you can use them in an IxLoad test.

There are two ways to configure a test in IxLoad with AWS instances, depending on how you want to configure the MAC and IP addresses for the test:

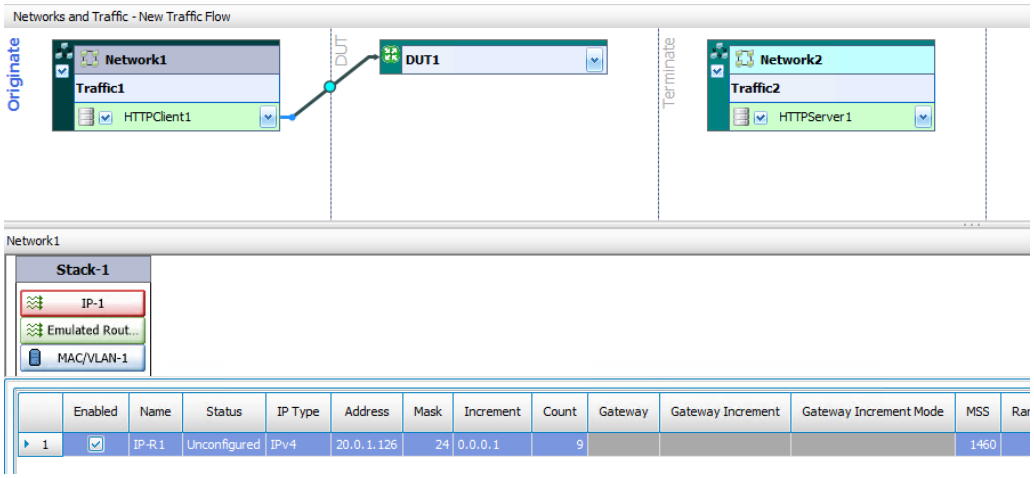
- Manual configuration, in which you enter the MAC and IP addresses
- Automatic configuration, in which IxLoad detects and learns the MAC and IP addresses

Manually Configuring MAC and IP Addresses

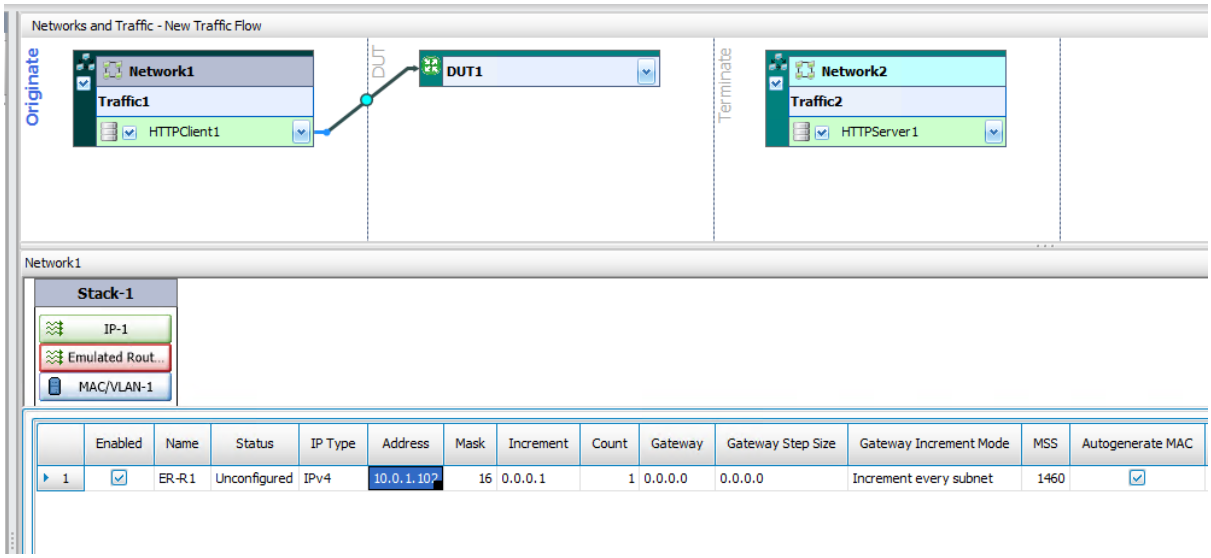
To manually configure an IxLoad test in AWS:

1. Start IxLoad.
2. Display the network configuration page, and add:
   one client NetTraffic
   one DUT NetTraffic
   one server NetTraffic

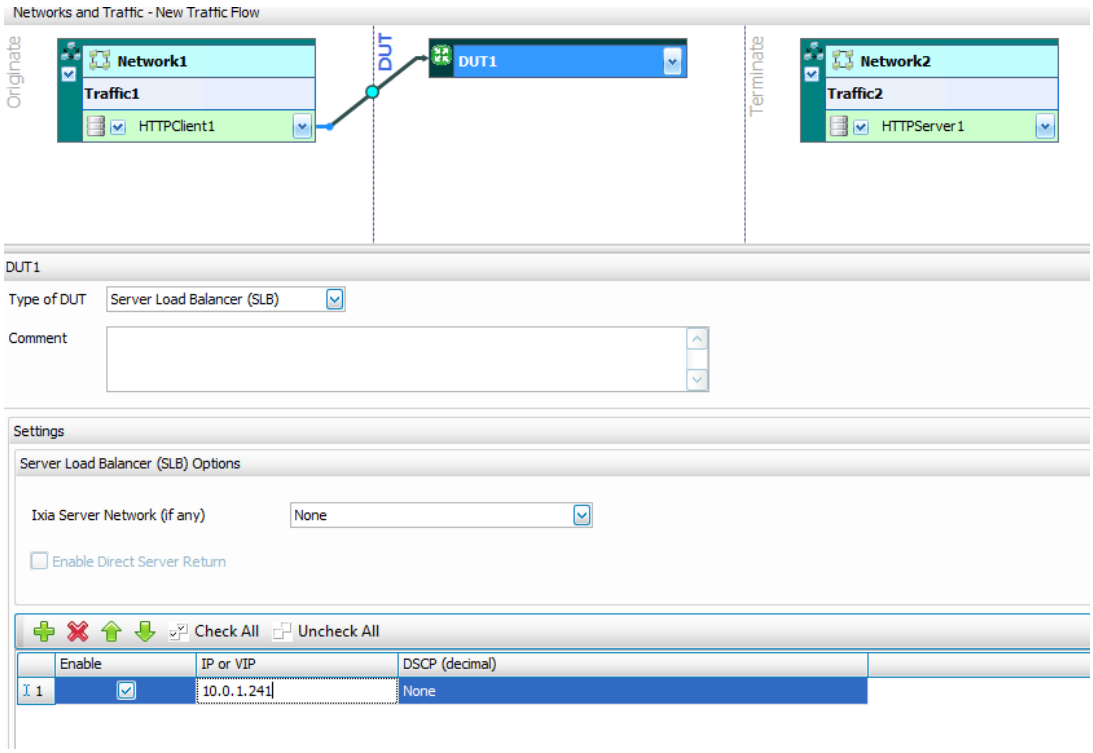
3. On the client NetTraffic, select the IP stack, and specify test port IP address for client traffic.



   On the client-side NetTraffic, select the Emulated Router stack, and specify its IP address. Use the address that is configured on the Eth1 interface of the first VMone instance. You can find this address under `VMONEINSTANCEDETAILS` on the Outputs tab of the CloudFormation | Stacks page.

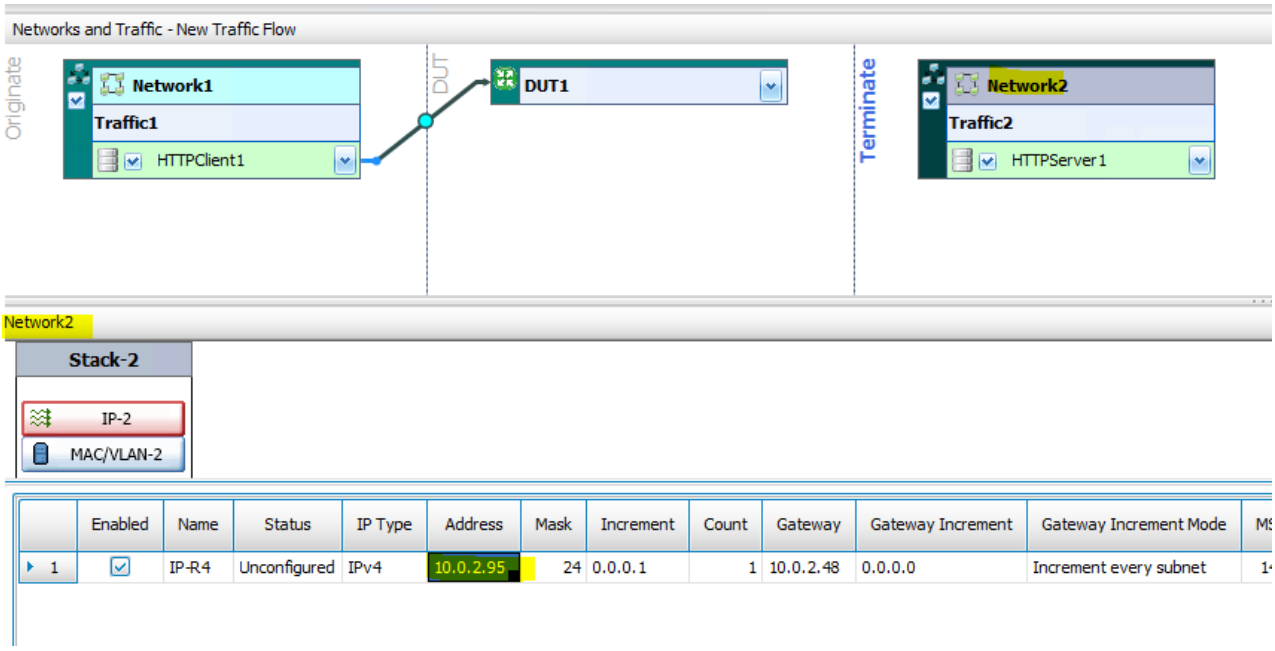


4. Select the DUT config, and specify IP address. Use the address that is assigned to Eth1 IP of the DUT deployed by the CFT. You can find this address under `DUTINSTANCEDETAILS` on the Outputs tab of the CloudFormation | Stacks page.



5. On the server NetTraffic, select the IP stack, and specify its IP address which is assigned to Eth2 of the 2nd VMone instance. For the gateway IP address, specify the address that is assigned to Eth2 IP of the DUT. You can find this address under `VMONEINSTANCEDETAILS` on the Outputs tab of

the CloudFormation | Stacks page.



6. Click **Start** to start the test.

Automatically Configuring MAC and IP Addresses

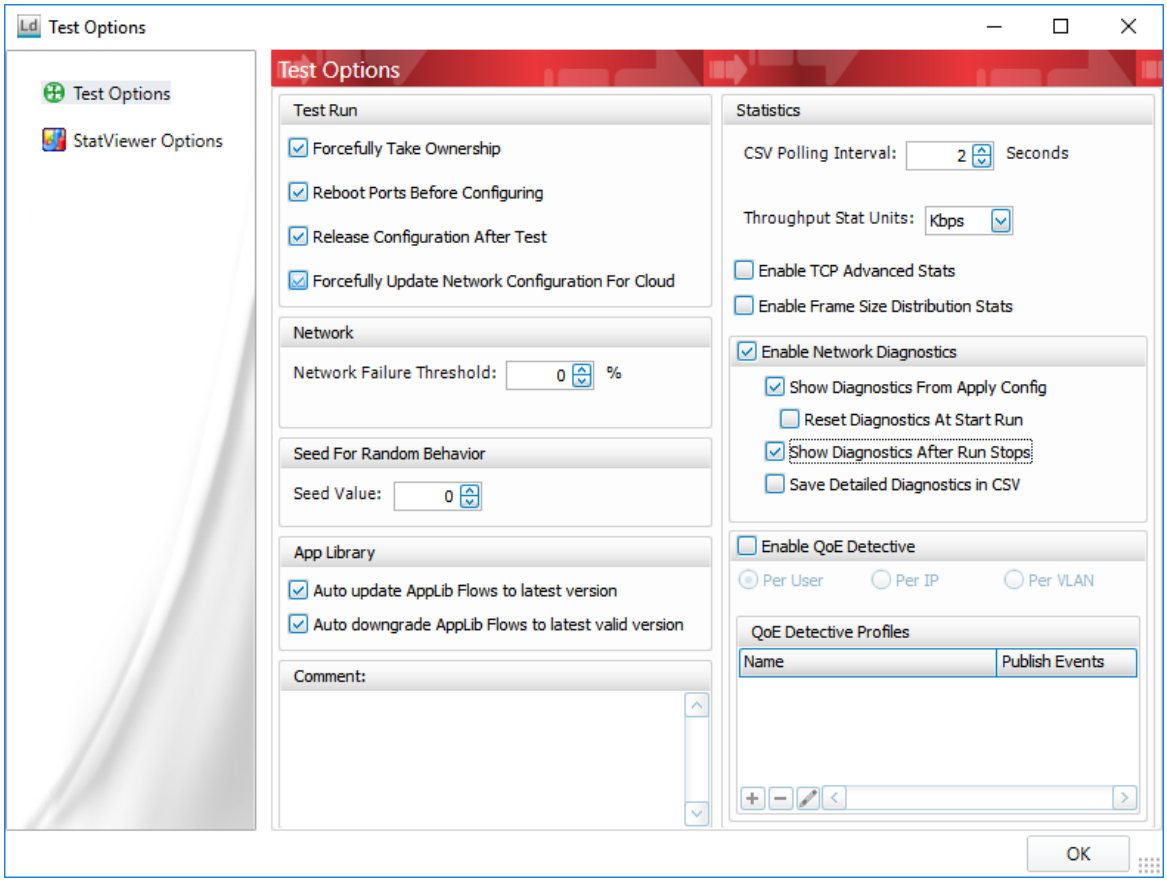
Before you begin

Before you begin to configure a cloud test with automatic addresses, there are some aspects of IxLoad's behavior in a cloud configuration that you should be aware of:

- The **Forcefully Update Network Configuration for Cloud** option
- The **Update Network For Cloud** button
- How IxLoad distributes the automatically-learned addresses from cloud instances

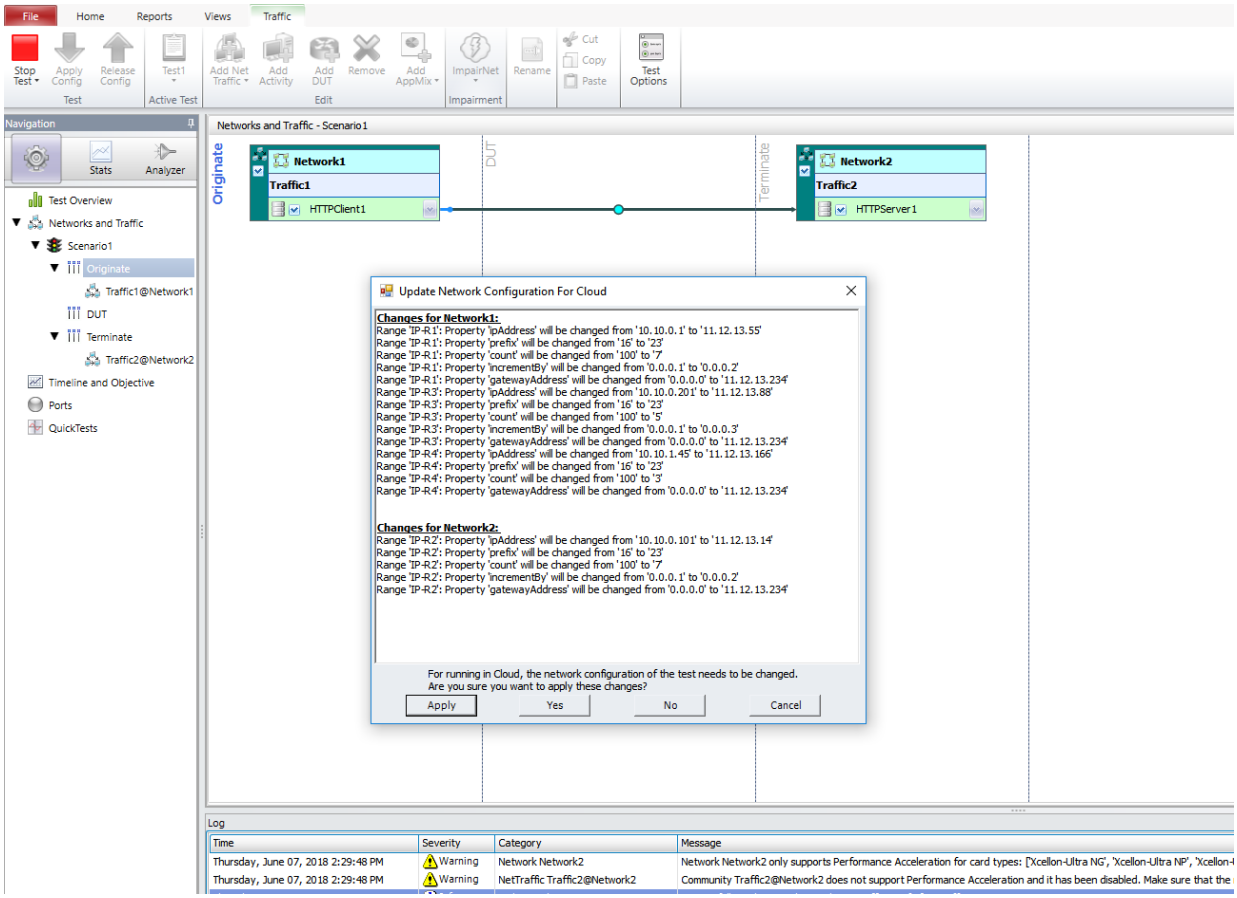
These are described in this section

On the **Test Run | Test Options** window, there is a checkbox named **Forcefully Update Network Configuration for Cloud**.



Checking this control allows IxLoad to update the test configuration with information from the cloud instances. During the Apply Config phase of the test (when IxLoad is downloading the test configuration to the ports), the following happens:

When running a test from the GUI	When running a test from an automation script:
If Forcefully Update is enabled, IxLoad prompts you if the configuration should be automatically changed. The prompt contains information about the changes that are going to be made **Yes:** If you choose "Yes", IxLoad will attempt to modify the configuration and then apply the configuration to the ports. **No:** If you choose "No", IxLoad will attempt to run with the existing configuration. **Cancel:** If you choose "Cancel", the test stops (it does not attempt to apply the configuration) **Apply:** If you choose "Apply", IxLoad applies the automatic changes, but does not start the test. If Forcefully Update is disabled, IxLoad will attempt to run with the existing configuration, and any errors will be displayed during the Apply Config phase.	If Forcefully Update is enabled, IxLoad attempts to modify the configuration and then apply the configuration to the ports. If Forcefully Update is disabled, IxLoad attempts to run with the existing configuration and any errors will be returned during the Apply Config phase.



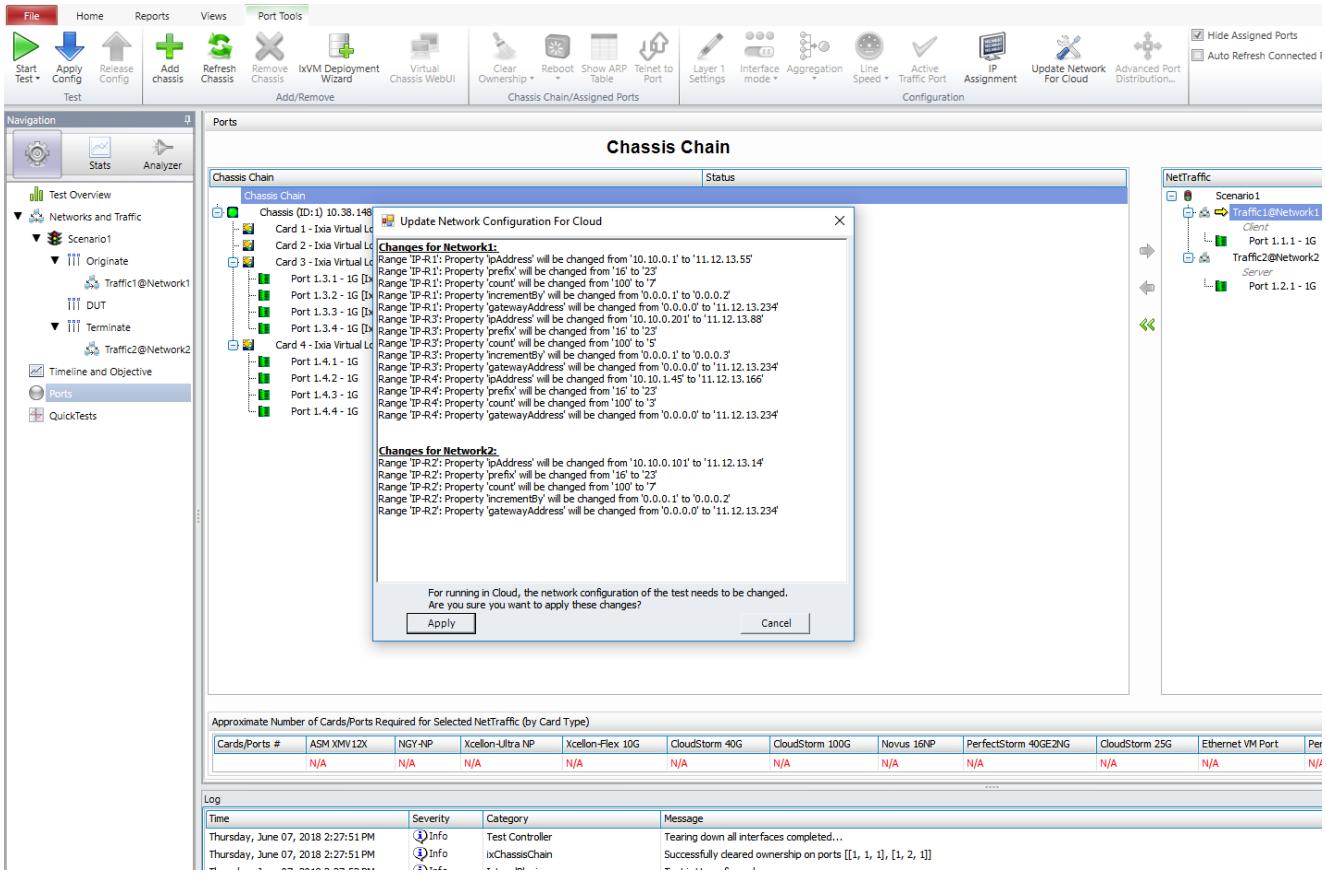
Update Network for Cloud Button

On the Ribbon (on the Network tab and Port Tools tab, next to IP Assignments) there is button named Update Network For Cloud.

You can click Update Network For Cloud any time when the test is not running, and IxLoad will retrieve the addresses available from the cloud instances.

Apply: If you choose “Apply”, IxLoad applies the automatic changes, but does not start the test.

Cancel: If you choose “Cancel”, the test stops (it does not attempt to apply the configuration).



How IxLoad updates the Network Configuration

With an Emulated Router

IxLoad uses the first allowed IP address on each assigned port as the Connected IP address. The number of ranges of Emulated Router and the count of each range are not important.

After the addresses have been learned and the configuration updated, the number of Emulated Router ranges will equal the number of ports assigned and count=1 on each range. If there are not enough ranges, extra ranges are added. If there are more ranges than ports assigned, the extra ranges are removed.

The 'Autogenerate MAC' option is disabled on all the Emulated Router ranges.

MAC addresses are updated.

If the test has both IPv4 and IPv6 addresses. IxLoad does not update the MAC addresses to avoid a MAC overlap error. The test will work because promiscuous mode is disabled.

VLAN is disabled on all ranges.

Unconnected IP addresses are not changed.

Without an Emulated Router

The Random IP checkbox is automatically disabled during the update.

The 'Autogenerate MAC' option is left as it is on all the IP ranges.

IxLoad uses the first chunk of IPs with equal increment on each port.

For example if the cloud allows the following IP addresses:
1.1.1.11
1.1.1.12
1.1.1.13
1.1.1.16
1.1.1.17
1.1.1.20
1.1.1.22
1.1.1.24
the IPs that IxLoad is going to use are startIP = 1.1.1.11, increment = 0.0.0.1, count = 3.

If the existing configuration has three ranges, IxLoad will use:

Range 1: startIP = 1.1.1.11, increment = 0.0.0.1, count = 3
Range 2: startIP = 1.1.1.16, increment = 0.0.0.1, count = 2
Range 3: startIP = 1.1.1.20, increment = 0.0.0.2, count = 3

IxLoad does not add or remove any IP ranges from the configuration. This means that some issues could occur.

For example, if multiple ports are assigned:

- If the per-port distribution type is IP Round Robin, the automatic update fails.
- If the per-port distribution type is Consecutive IPs, the automatic update fails if the existing ranges don't have an equal number of IPs.

If the per-port distribution type is Consecutive Ranges, there shouldn't be any issues, except the case when there are too many ranges in the original configuration.

The typical sequence that can come from Cloud is a counter or a list of counters. A counter is a sequence that can be described with start, increment and count.

Counters and ranges

In the example above, there is a list of 3 counters:

Counter1: start=1.1.1.11, increment=0.0.0.1, count=3
Counter2: start=1.1.1.16, increment=0.0.0.1, count=2
Counter3: start=1.1.1.20, increment=0.0.0.2, count=3

What happens if there are fewer counters than ranges?

Using the example above, assume that there are 4 ranges, and the total number of allowed IPs is 8.

First, the number of IPs is divided by the number of ranges: 8/4=2

This means that IxLoad first tries to allocate 2 IPs for each range.

If it fails, IxLoad tries again with 1 IP for each range.

IxLoad tries allocating 2 IPs for each range by taking IPs from each counter in chunks of 2.

If a counter remains with less than 2 IPs, IxLoad moves to the next range.

Therefore, using the example addresses above, IxLoad attempts to distribute the addresses as follows:

1.1.1.11 1.1.1.12	Range 1
1.1.1.13	skipped, moves to the next counter
1.1.1.16 1.1.1.17	Range 2
1.1.1.20 1.1.1.22	Range 3
1.1.1.24	skipped

At this pont, it gives up because there are no counters to continue with and cover the fourth range. This means that the attempt failed and it will try with 1 IP per range.

With 1 IP per range, it will allocate the addresses as follows:

1.1.1.11	Range 1
1.1.1.12	Range 2
1.1.1.13	Rrange 3
1.1.1.16	Range 4

Suppose there are 3 ranges instead of 4.

The number of IPs is divided by the number of ranges: 8/3=2

It tries with 2 IPs per range, and it succeeds:

1.1.1.11 and 1.1.1.12	Range 1
1.1.1.16 and 1.1.1.17	Range 2
1.1.1.20 and 1.1.1.22	Range 3

What happens if the sequence is not a counter or a list of counters?

There are situations in which the sequence may not be a counter or a list of counters

For example, this sequence:

```
1.1.1.1
1.1.1.2
1.1.1.3
1.1.2.1
1.1.2.2
1.1.1.3
1.1.3.1
1.1.3.2
1.1.3.3 …
```

In these situations, IxLoad simply converts this sequence to a list of counters of 1 element each and applies the usual algorithm. Each range will receive a single IP.

What happens if a range is OK and doesn't need to be changed?

- All the IP addresses must be included in the sequence of the allowed IPs
- The prefix must be correct
- The gateway must be correct
- Random IP must be disabled

If the cloud allows the following sequence of IPs:

```
1.1.1.11
1.1.1.12
1.1.1.13
1.1.1.16
1.1.1.17
1.1.1.20
1.1.1.22
1.1.1.24
```

and the configuration has 3 ranges,

One range has IP = 1.1.1.12, increment = 0.0.0.4, count = 3

The other two ranges have bad configurations because they that do not match.

The good range will not be changed.

IP addresses 1.1.1.12, 1.1.1.16 and 1.1.1.20 will be removed from the sequence of allowed IPs as they are already used by this range:

1.1.1.11
~~1.1.1.12~~
1.1.1.13
~~1.1.1.16~~
1.1.1.17
~~1.1.1.20~~
1.1.1.22
1.1.1.24

The remaining sequence will be represented as 3 counters:

Counter1: start=1.1.1.11, increment=0.0.0.2, count=2
Counter2: start=1.1.1.17, increment=0.0.0.5, count=2
Counter3: start=1.1.1.24, increment=0.0.0.0, count=1

Then the usual algorithm will be applied for the two remaining ranges:

Range1 will receive addresses 1.1.1.11 and 1.1.1.13
Range2 will receive addresses 1.1.1.17 and 1.1.1.22

Limitations

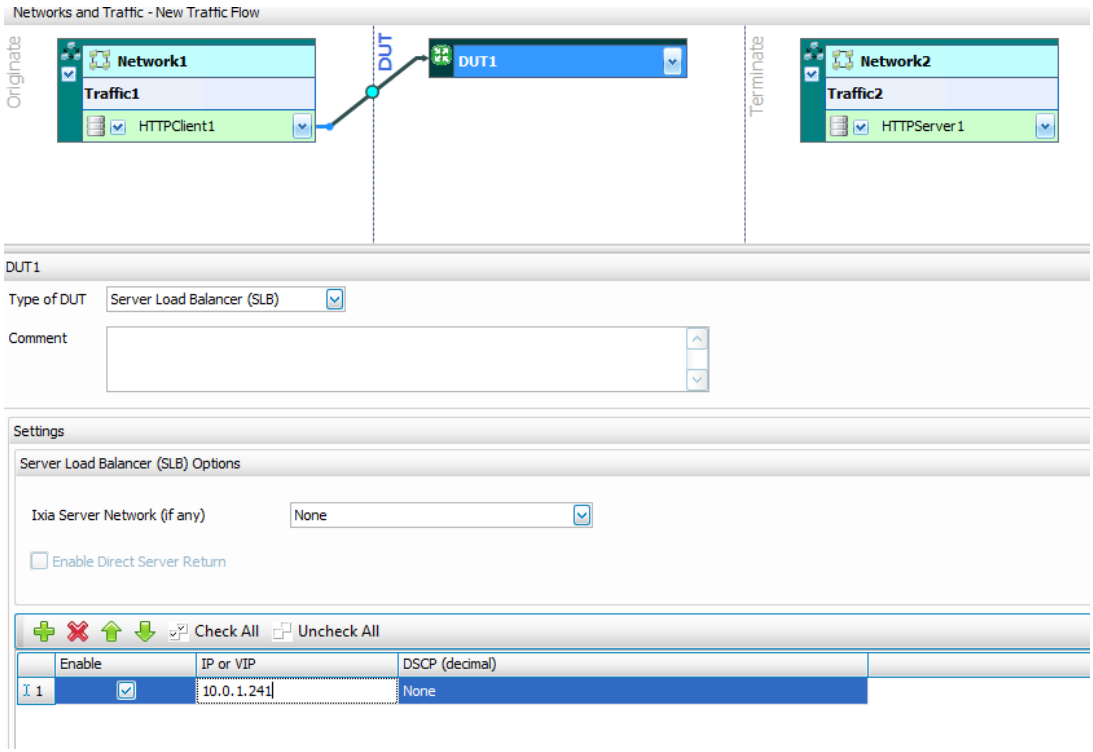
Automatic address learning has the following limitations:

- Only plain IP stacks (with or without emulated router) are updated.
- For Custom Mesh configurations, updates are not reflected in the Custom Mesh GUI.
- Only IP addresses in Stack Manager stacks are updated. IP addresses used in other areas of IxLoad (for example, as the destination for L4-7 traffic or on a virtual DUT) are not updated. These addresses must be updated manually.
- If the configuration has IPv6 enabled on any range but the port has no IPv6 addresses, the update will fail. The reverse is also true: if the port has only IPv6 addresses but IPv6 is disabled on the ranges, the update will also fail.

Configuring a test with automatic MAC and IP address learning

To automatically configure an IxLoad test in AWS:

1. Start IxLoad.
2. On the Test Run | Test Options window, check **Forcefully Update Network Configuration for Cloud** option.
3. Display the network configuration page, and add:
   one client NetTraffic
   one DUT NetTraffic
   one server NetTraffic
4. Select the DUT config, and specify IP address.

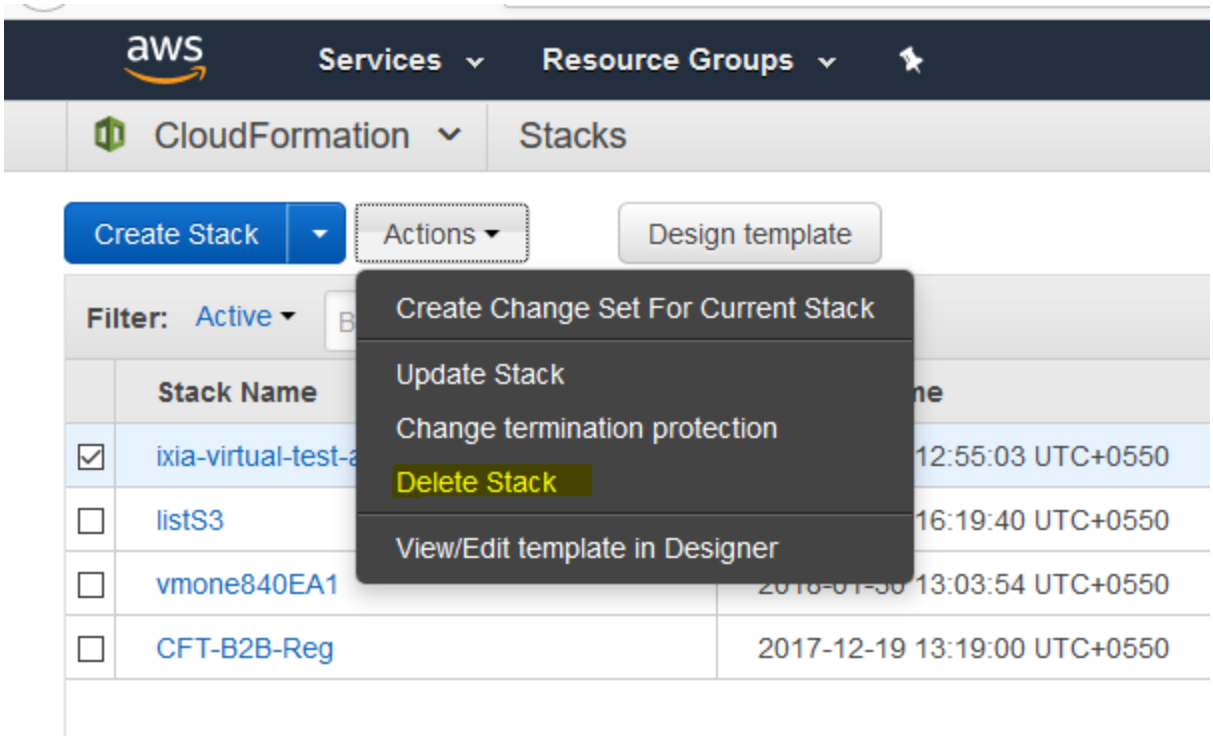


5. On the Ribbon (on the Network tab and Port Tools tab, next to IP Assignments), click **Update Network For Cloud**.
6. Click **Start** to start the test.

Deleting a stack

To delete a stack:

1. Display the CloudFormation page.
2. Select the stack
3. Click the "Actions" drop down list, then click Delete Stack.
   AWS terminates all instances, and deletes all resources created during the stack creation process.



Known Limitations

Delete stack operations can take more than 10 minutes

Instead of an elastic IP address, a public IP address is assigned to management interface of each instance

A maximum of 5 elastic IP addresses can be allocated per account / per region

A maximum of 5 VPCs can be created per account / per region

For instances based on the c3.8xlarge, c4.8xlarge, and similar instance types, a maximum of 8 interfaces can be created on each instance.

For instances based on the c3.8xlarge, c4.8xlarge, and similar instance types, a maximum of 30 IP addresses (including secondary IPs) can be allocated per interface on each instance.

Promiscuous mode must be disabled on all interfaces. All IP packets will use the interface MAC address

All Data protocols are supported on IxLoad AWS instances. Storage protocols are not supported.

The IP versions (IPv4/IPv6) used on the VM port and in the IxLoad configuration must be compatible:

- If the configuration has IPv6 enabled on any range, but the port has no IPv6 addresses, the update will fail. If the port has only IPv6 addresses but the configuration has no IPv6 ranges enabled, the update will fail.
- If the configuration has IPv4 enabled on any range, but the port has no IPv4 addresses, the update will fail. If the port has only IPv4 addresses, but the configuration has no IPv4 addresses ranges enabled, the update will fail.

Ixia, a Keysight Business
26601 West Agoura Road
Calabasas, California 91302