Kony MobileFabric™

Kony MobileFabric QuickStart Guide Series

Getting Started with Amazon AWS and Kony MobileFabric

Release 6.5

Revision History

Date	Document Version	Description of Modifications/Release
09/24/2015	1.2	Document Released for Amazon AWS Kony MobileFabric 6.5.2
07/09/2015	1.0	Document Released for Amazon AWS Kony MobileFabric 6.5

Table of Contents

1. Overview

Kony MobileFabric is a leading Mobile Backend as a Service (MBaaS) that developers can leverage from any app development platform they choose using the various MobileFabric SDKs. It provides powerful API Management and Analytics capabilities along with the following set of configurable enterprise services to simplify your mobile backend and significantly reduce the time it takes to develop mobile applications.

Kony MobileFabric provides enterprise security and complex system integration services and allows developers to focus on building app experiences. This is accomplished by providing a powerful set of services to handle identity, integration, orchestration, data sync, and messaging. When these services are configured within Kony MobileFabric, they can easily be incorporated into a mobile application using any third-party app development tool using our SDKs or direct REST API interface.

MobileFabric has multiple features that can be used - Identity, Integration, Orchestration, Sync, and Messaging. These features can be accessed via a common, centralized console.

For successful authentication with users, and to access centralized features of MobileFabric, Kony recommends that you install the following MobileFabric features on premises:

- Kony MobileFabric Identity and Console
- Kony MobileFabric Integration
- Kony MobileFabric Messaging
- Kony MobileFabric Sync

The following are the five major services offered by Kony MobileFabric:

- **Identity**

  Protect your enterprise data by using your existing identity infrastructure to authenticate and authorize app users. Integrate with Active Directory, Salesforce, SAP, Kony's Cloud User Repository or other standardized identity providers via OAuth or SAML.

- **Integration**

  Easily mobilize your enterprise data such as Salesforce and SAP using business adaptors, or any system that provides standard REST, SOAP, XML or JSON services. These services can be further enhanced for mobile using powerful transformation and caching.

- **Service Orchestration**

  Optimize the user experience by creating workflows and composite services including custom logic and data processing on the server side to reduce the workload on the device.

- **Messaging Notification**

  Provide intelligent, real-time messaging by delivering targeted notifications and alerts based on user information and location to all device types. Configure automated campaigns or events you can trigger to engage with users over Push, SMS, Email or iOS Passbook.

- **Data synchronization**

  Build offline apps that synchronize relational data between the device and your existing back-end systems. Configure server-side rules and policies to control data access, filtering and automatic conflict resolution.

1.1 MobileFabric AWS Editions and Capacity

Kony MobileFabric is offered on AWS in two separate editions:

- **Kony MobileFabric Developer Edition** is a free offering available for development, testing or small production workloads. Typically the developer edition can support an app that has up to 20,000 unique user sessions (unique visits) per month.

- **Kony MobileFabric Express Edition** is a paid offering on a larger EC2 server that typically supports up to 75,000 user sessions.

There is no built-in limit to the number of users or user sessions that each edition can support beyond what can be achieved using the memory, CPU and network capabilities of the EC2 server size. For larger scale deployments in a high availability configuration, Kony MobileFabric Standard, Professional and Enterprise editions are available as a SaaS offering on Kony Cloud or available on-premise.

2. AWS Marketplace and Kony MobileFabric

Amazon AWS Marketplace is an online store that helps Amazon customers find, buy, and immediately start using the software and services they need to build products and run their businesses. If you already have an AWS account, follow these steps to create an MobileFabric instance and launch it:

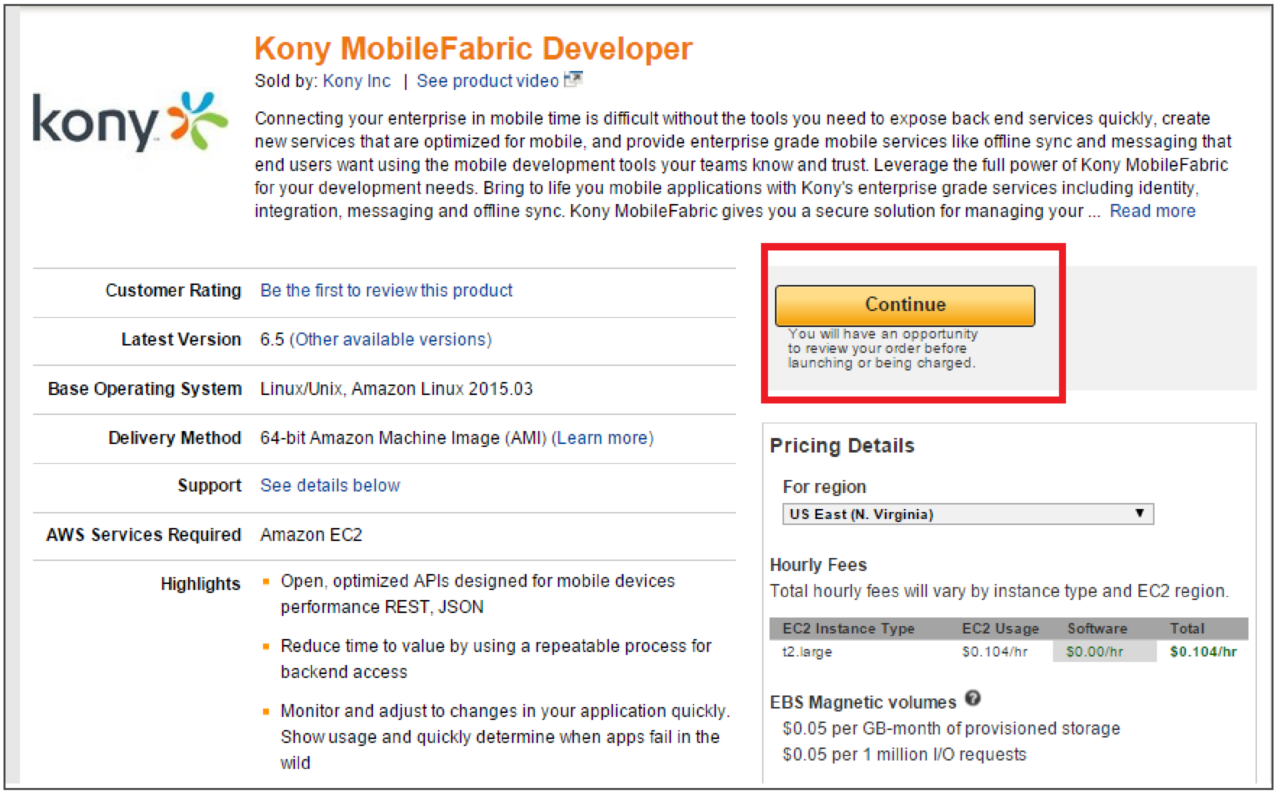
1. Configuring Kony MobileFabric as required for Developer or Express version:
    - Configuring Kony MobileFabric Developer 1-Click Launch
    - Configuring Kony MobileFabric Developer Manual Launch
    - Configuring Kony MobileFabric Express 1-Click Launch
    - Configuring Kony MobileFabric Express Manual Launch
2. Launching the MobileFabric Instance

2.1 Configuring AWS Kony MobileFabric Developer: 1-Click Launch

To 1-Click launch Kony MobileFabric Developer Edition , follow these steps:

1. Go to https://aws.amazon.com/marketplace.
2. Sign in to your AWS Account.
3. Browse to the Kony MobileFabric AWS product page.

4. Click **Continue**.



Kony MobileFabric Developer

Sold by: Kony Inc | See product video

Connecting your enterprise in mobile time is difficult without the tools you need to expose back end services quickly, create new services that are optimized for mobile, and provide enterprise grade mobile services like offline sync and messaging that end users want using the mobile development tools your teams know and trust. Leverage the full power of Kony MobileFabric for your development needs. Bring to life you mobile applications with Kony's enterprise grade services including identity, integration, messaging and offline sync. Kony MobileFabric gives you a secure solution for managing your ... Read more

Customer Rating	Be the first to review this product
Latest Version	6.5 (Other available versions)
Base Operating System	Linux/Unix, Amazon Linux 2015.03
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	See details below
AWS Services Required	Amazon EC2
Highlights	- Open, optimized APIs designed for mobile devices performance REST, JSON - Reduce time to value by using a repeatable process for backend access - Monitor and adjust to changes in your application quickly. Show usage and quickly determine when apps fail in the wild

You will have an opportunity to review your order before launching or being charged.

Pricing Details

For region

US East (N. Virginia)

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

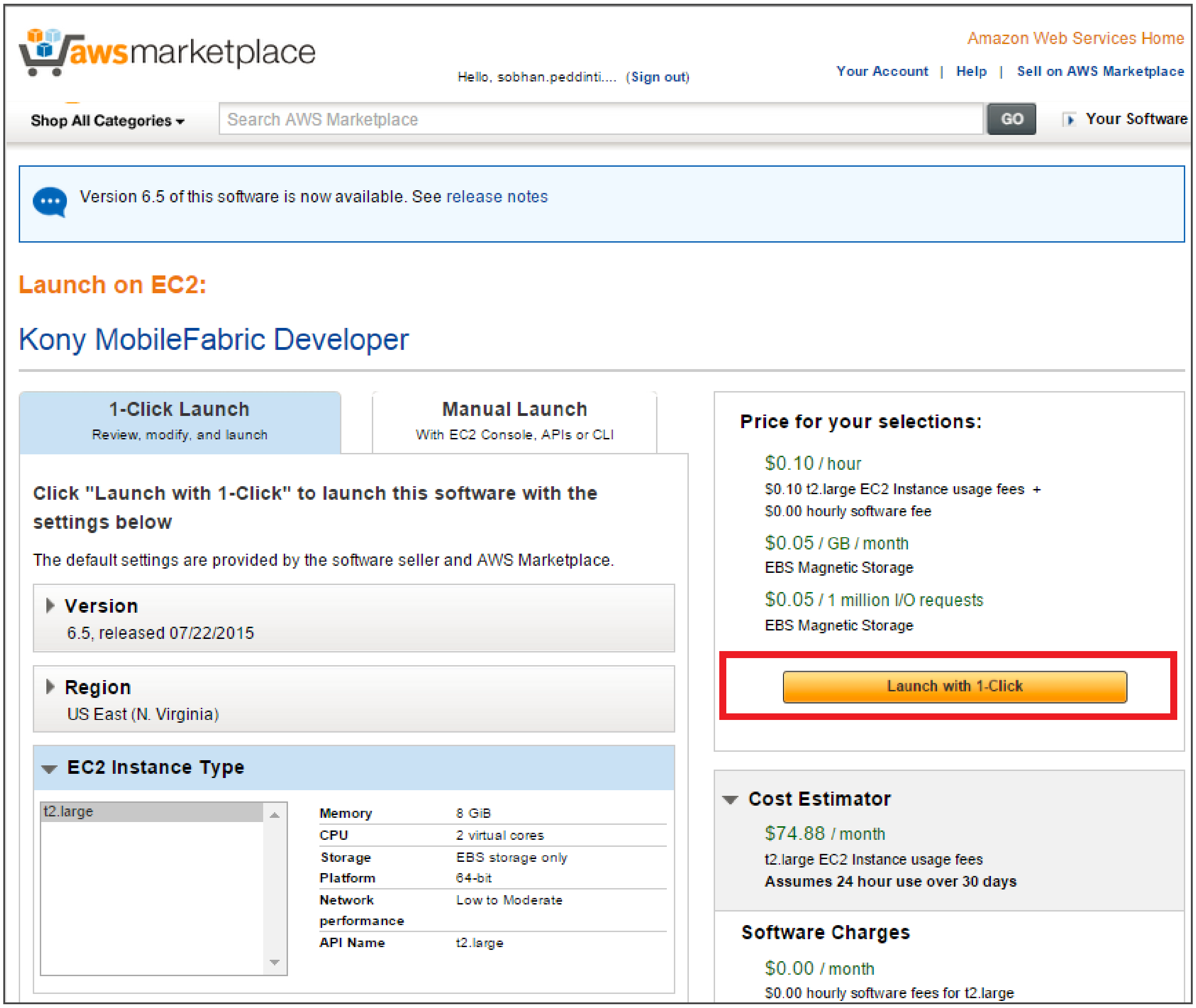
EC2 Instance Type	EC2 Usage	Software	Total
t2.large	$0.104/hr	$0.00/hr	$0.104/hr

EBS Magnetic volumes

$0.05 per GB-month of provisioned storage

$0.05 per 1 million I/O requests

5. The **Launch on EC2** page appears. Under the **Kony MobileFabric Developer**, click the **1-Click Launch** tab:


awsmarketplace
Amazon Web Services Home
Hello, sobhan.peddinti.... (Sign out)
Your Account | Help | Sell on AWS Marketplace
Shop All Categories
Search AWS Marketplace
GO
Your Software

Version 6.5 of this software is now available. See release notes

Launch on EC2:

Kony MobileFabric Developer

1-Click Launch
Review, modify, and launch
Manual Launch
With EC2 Console, APIs or CLI

Click "Launch with 1-Click" to launch this software with the settings below

The default settings are provided by the software seller and AWS Marketplace.

Version
6.5, released 07/22/2015

Region
US East (N. Virginia)

EC2 Instance Type

t2.large

Memory	8 GiB
CPU	2 virtual cores
Storage	EBS storage only
Platform	64-bit
Network performance	Low to Moderate
API Name	t2.large

Price for your selections:

$0.10 / hour
$0.10 t2.large EC2 Instance usage fees +
$0.00 hourly software fee

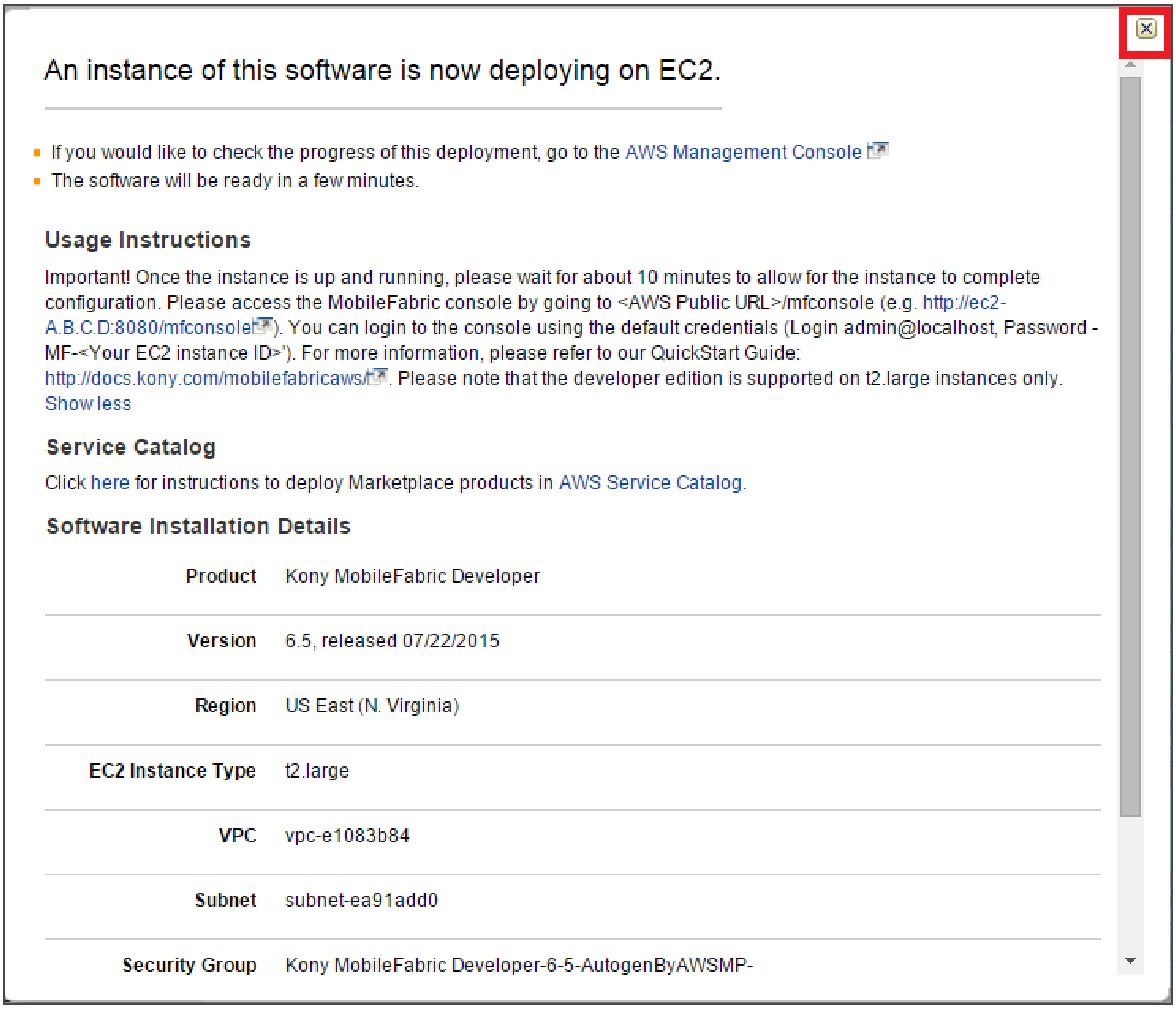
$0.05 / GB / month
EBS Magnetic Storage

$0.05 / 1 million I/O requests
EBS Magnetic Storage

Cost Estimator
$74.88 / month
t2.large EC2 Instance usage fees
Assumes 24 hour use over 30 days

Software Charges
$0.00 / month
$0.00 hourly software fees for t2.large

6. The Instance details pop-up appears. Verify that instance information is correct and click the **Cancel** button to close the window.



An instance of this software is now deploying on EC2.

- If you would like to check the progress of this deployment, go to the AWS Management Console
- The software will be ready in a few minutes.

Usage Instructions

Important! Once the instance is up and running, please wait for about 10 minutes to allow for the instance to complete configuration. Please access the MobileFabric console by going to <AWS Public URL>/mfconsole (e.g. http://ec2-A.B.C.D:8080/mfconsole). You can login to the console using the default credentials (Login admin@localhost, Password - MF-<Your EC2 instance ID>'). For more information, please refer to our QuickStart Guide: http://docs.kony.com/mobilefabricaws/. Please note that the developer edition is supported on t2.large instances only.
Show less

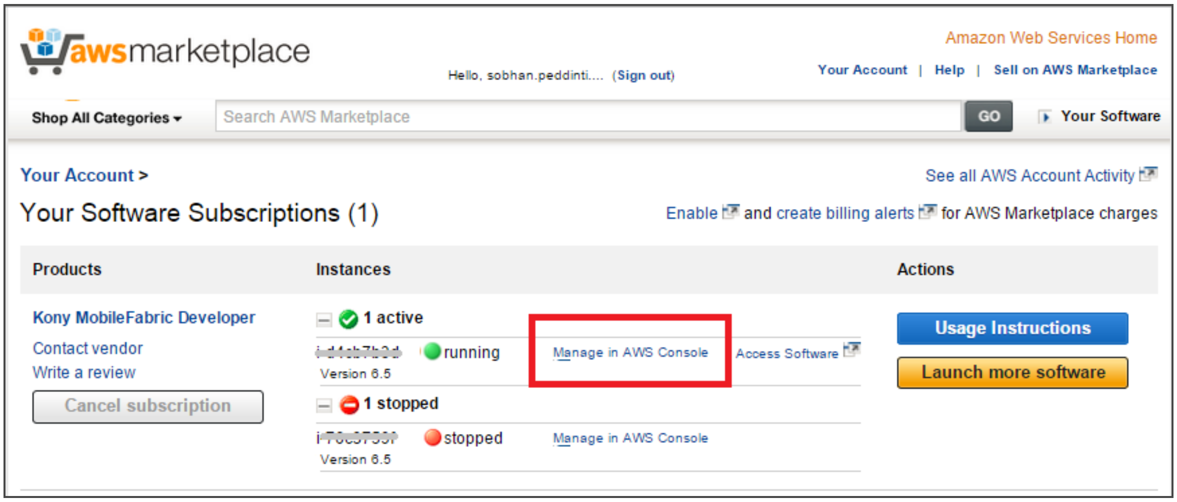
Service Catalog

Click here for instructions to deploy Marketplace products in AWS Service Catalog.

Software Installation Details

Product	Kony MobileFabric Developer
Version	6.5, released 07/22/2015
Region	US East (N. Virginia)
EC2 Instance Type	t2.large
VPC	vpc-e1083b84
Subnet	subnet-ea91add0
Security Group	Kony MobileFabric Developer-6-5-AutogenByAWSMP-

7. In the page that appears, click the **Manage in AWS Console** link to manage your instances, shown below:



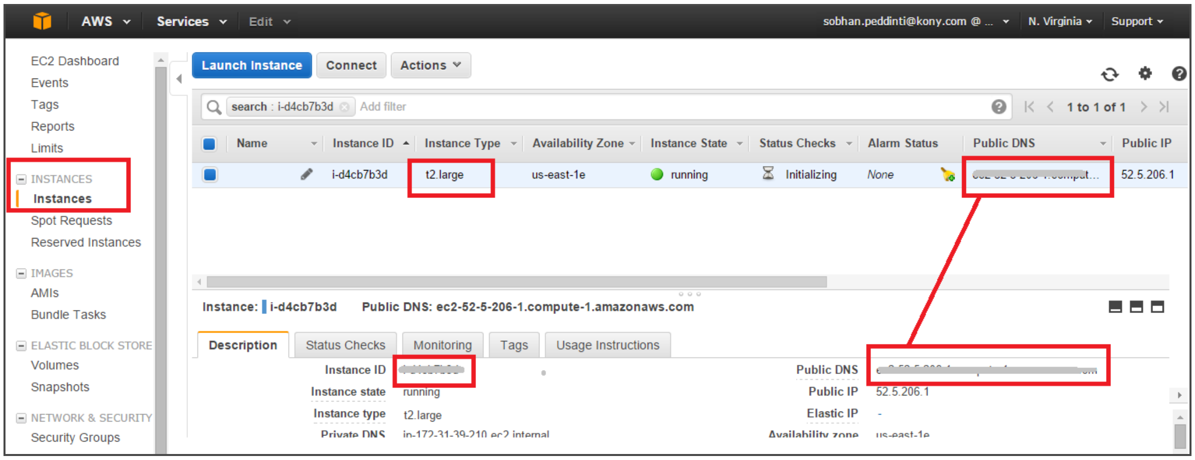
awsmarketplace
Amazon Web Services Home
Hello, sobhan.peddinti.... (Sign out)
Your Account | Help | Sell on AWS Marketplace
Shop All Categories
Search AWS Marketplace
GO
Your Software
Your Account >
See all AWS Account Activity
Enable and create billing alerts for AWS Marketplace charges

Your Software Subscriptions (1)

Products	Instances				Actions
Kony MobileFabric Developer Contact vendor Write a review Cancel subscription	1 active				Usage Instructions Launch more software
	[redacted] Version 6.5	running	Manage in AWS Console	Access Software	
	1 stopped				
	[redacted] Version 6.5	stopped	Manage in AWS Console		

In the **Instances**, the MobileFabric instance is created, shown below:

- **Instance Type** indicates the instance type that you have chosen.
- **Public DNS** indicates URL for the MobileFabric Console.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-d4cb7b3d	t2.large	us-east-1e	running	Initializing	None	[redacted]	52.5.206.1

Instance: i-d4cb7b3d Public DNS: ec2-52-5-206-1.compute-1.amazonaws.com

Instance ID	[redacted]	Public DNS	[redacted]
Instance state	running	Public IP	52.5.206.1
Instance type	t2.large	Elastic IP	-
Private DNS	ip-172-31-39-210.ec2.internal	Availability zone	us-east-1e

You can now launch the MobileFabric Instance.

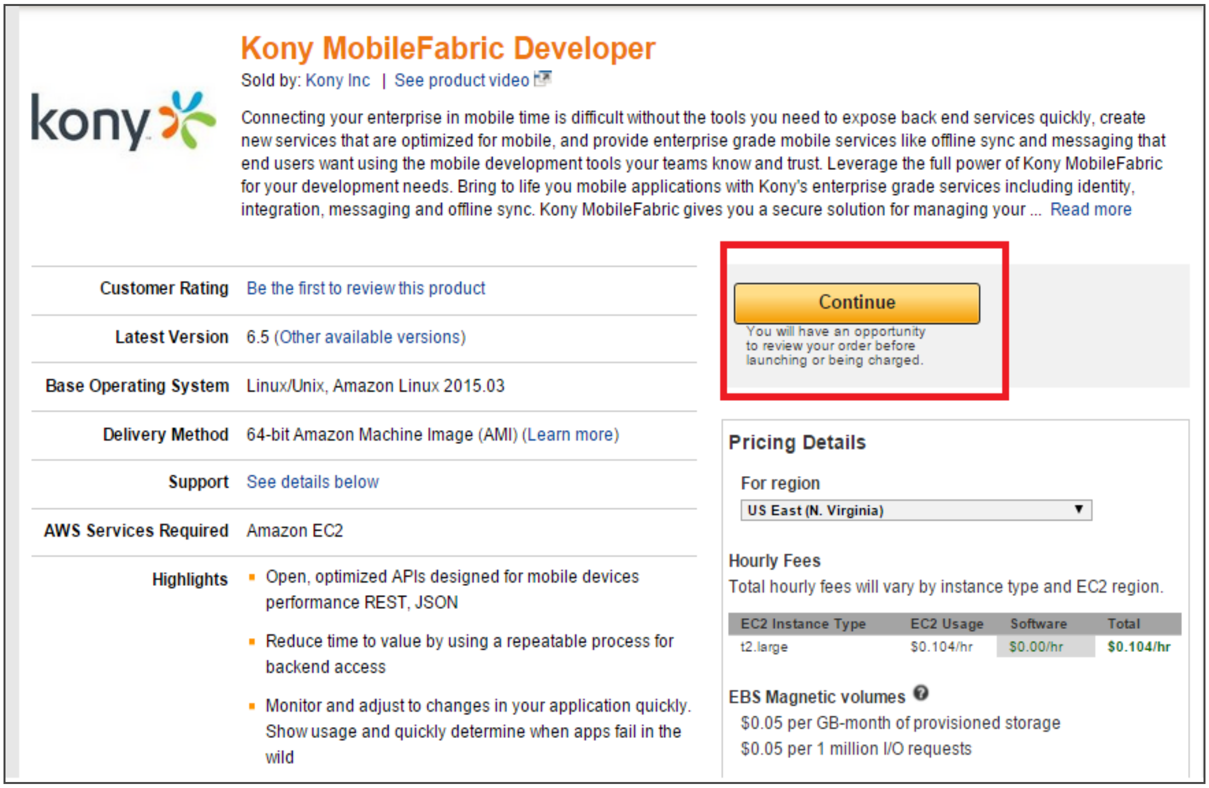
2.2 Configuring AWS Kony MobileFabric Developer: Manual Launch

To manually launch Kony MobileFabric Developer Edition, follow these steps:

1. Go to https://aws.amazon.com/marketplace.

2. Sign in to your AWS Account.

3. On the AWS Kony MobileFabric product page, click **Continue**.



Kony MobileFabric Developer

Sold by: Kony Inc | See product video

Connecting your enterprise in mobile time is difficult without the tools you need to expose back end services quickly, create new services that are optimized for mobile, and provide enterprise grade mobile services like offline sync and messaging that end users want using the mobile development tools your teams know and trust. Leverage the full power of Kony MobileFabric for your development needs. Bring to life you mobile applications with Kony's enterprise grade services including identity, integration, messaging and offline sync. Kony MobileFabric gives you a secure solution for managing your ... Read more

Customer Rating	Be the first to review this product
Latest Version	6.5 (Other available versions)
Base Operating System	Linux/Unix, Amazon Linux 2015.03
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	See details below
AWS Services Required	Amazon EC2
Highlights	▪ Open, optimized APIs designed for mobile devices performance REST, JSON ▪ Reduce time to value by using a repeatable process for backend access ▪ Monitor and adjust to changes in your application quickly. Show usage and quickly determine when apps fail in the wild

Continue

You will have an opportunity to review your order before launching or being charged.

Pricing Details

For region

US East (N. Virginia) ▼

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

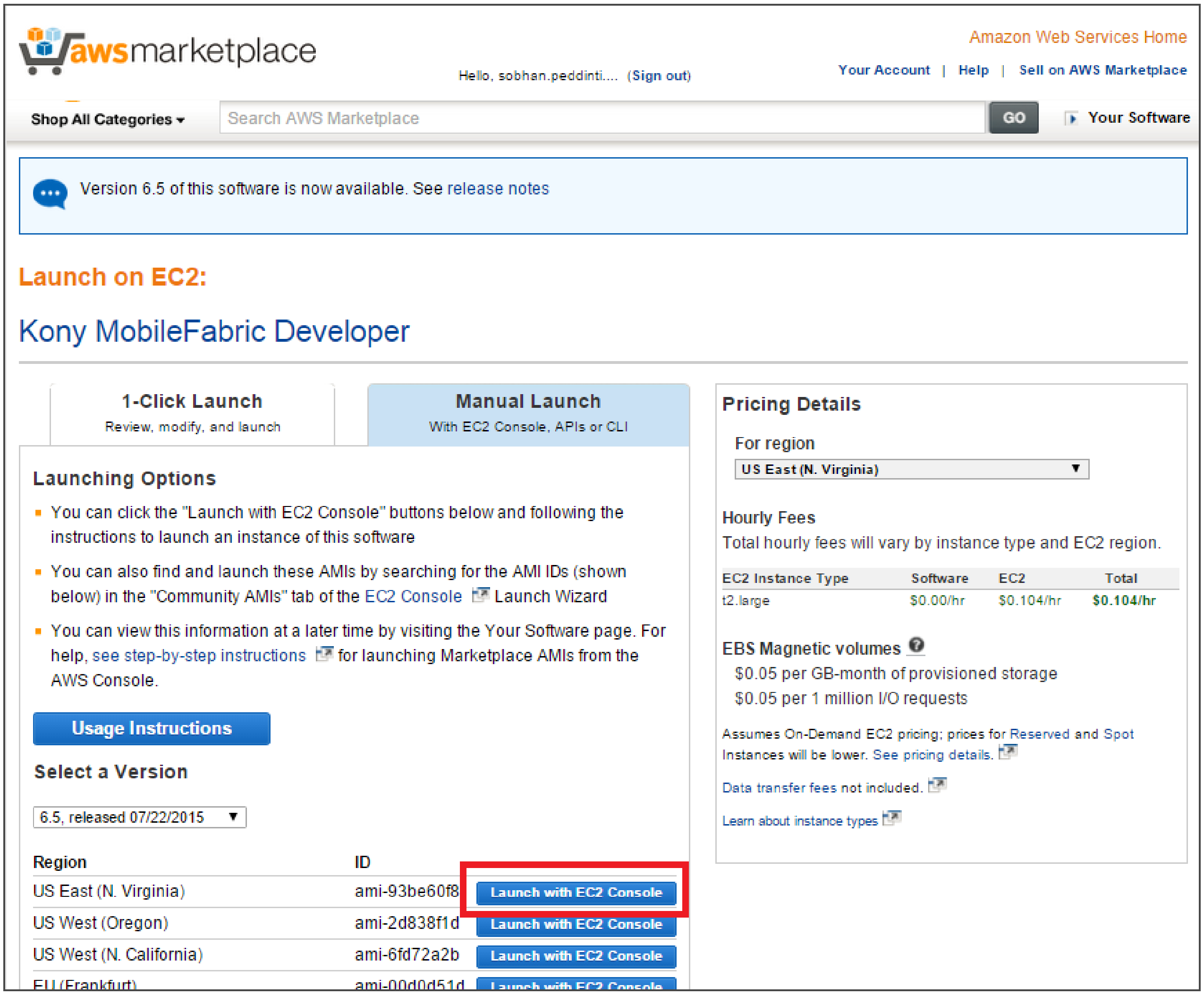
EC2 Instance Type	EC2 Usage	Software	Total
t2.large	$0.104/hr	$0.00/hr	$0.104/hr

EBS Magnetic volumes

$0.05 per GB-month of provisioned storage

$0.05 per 1 million I/O requests

4. The **Launch on EC2** page appears. Under the **Kony MobileFabric Developer**, click the **Manual Launch** tab.


awsmarketplace
Amazon Web Services Home
Hello, sobhan.peddinti.... (Sign out)
Your Account | Help | Sell on AWS Marketplace
Shop All Categories ▾
Search AWS Marketplace
GO
Your Software

Version 6.5 of this software is now available. See release notes

Launch on EC2:

Kony MobileFabric Developer

1-Click Launch
Review, modify, and launch
Manual Launch
With EC2 Console, APIs or CLI

Launching Options

- You can click the "Launch with EC2 Console" buttons below and following the instructions to launch an instance of this software
- You can also find and launch these AMIs by searching for the AMI IDs (shown below) in the "Community AMIs" tab of the EC2 Console Launch Wizard
- You can view this information at a later time by visiting the Your Software page. For help, see step-by-step instructions for launching Marketplace AMIs from the AWS Console.

Select a Version

6.5, released 07/22/2015 ▾

Region	ID	
US East (N. Virginia)	ami-93be60f8	Launch with EC2 Console
US West (Oregon)	ami-2d838f1d	Launch with EC2 Console
US West (N. California)	ami-6fd72a2b	Launch with EC2 Console
EU (Frankfurt)	ami-00d0d51d	Launch with EC2 Console

Pricing Details

For region

US East (N. Virginia) ▾

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

EC2 Instance Type	Software	EC2	Total
t2.large	$0.00/hr	$0.104/hr	$0.104/hr

EBS Magnetic volumes

$0.05 per GB-month of provisioned storage

$0.05 per 1 million I/O requests

Assumes On-Demand EC2 pricing; prices for Reserved and Spot Instances will be lower. See pricing details.

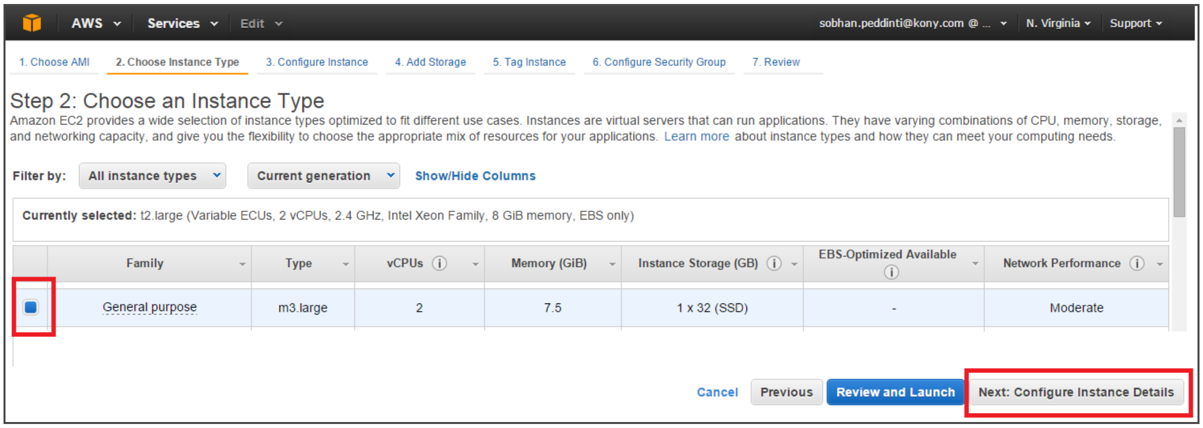
Data transfer fees not included.

Learn about instance types

5. In the **Manual Launch** tab, configure the required fields such as for region and version. In the **Region** section, click the **Launch with EC2 Console** button.

6. In the **Step 2: Choose an Instance Type** page, click the instance for type m3.large, shown below:

7. Click the **Next: Configure Instance Details** button.



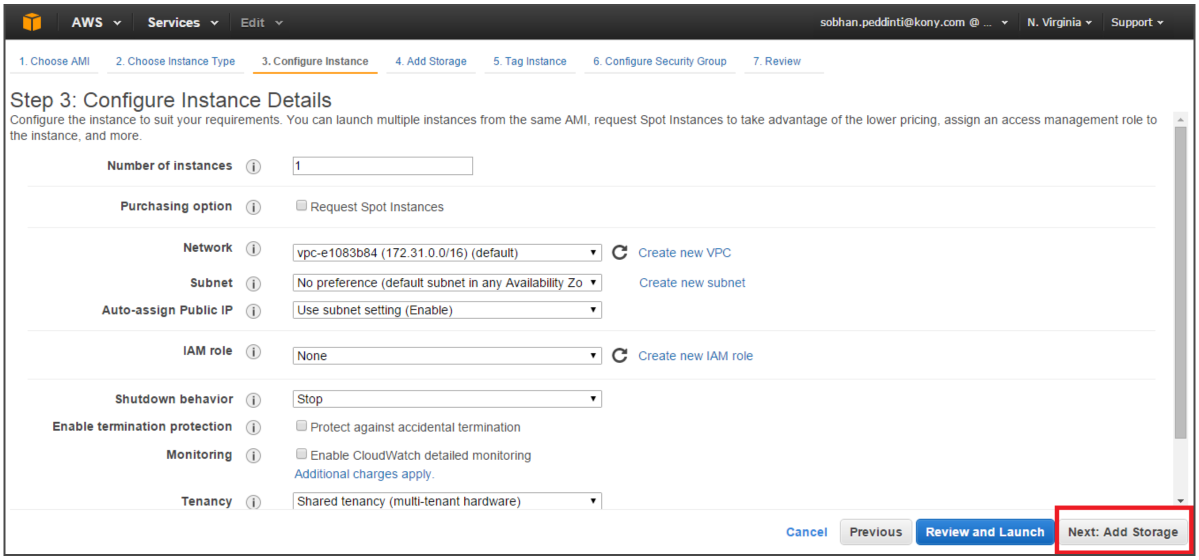
Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.4 GHz, Intel Xeon Family, 8 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
■	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate

8. In the **Step 3: Configure Instance Details** page, leave the defaults settings and click the **Next: Add Storage** button, shown below:



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-e1083b84 (172.31.0.0/16) (default)

Subnet: No preference (default subnet in any Availability Zo

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None

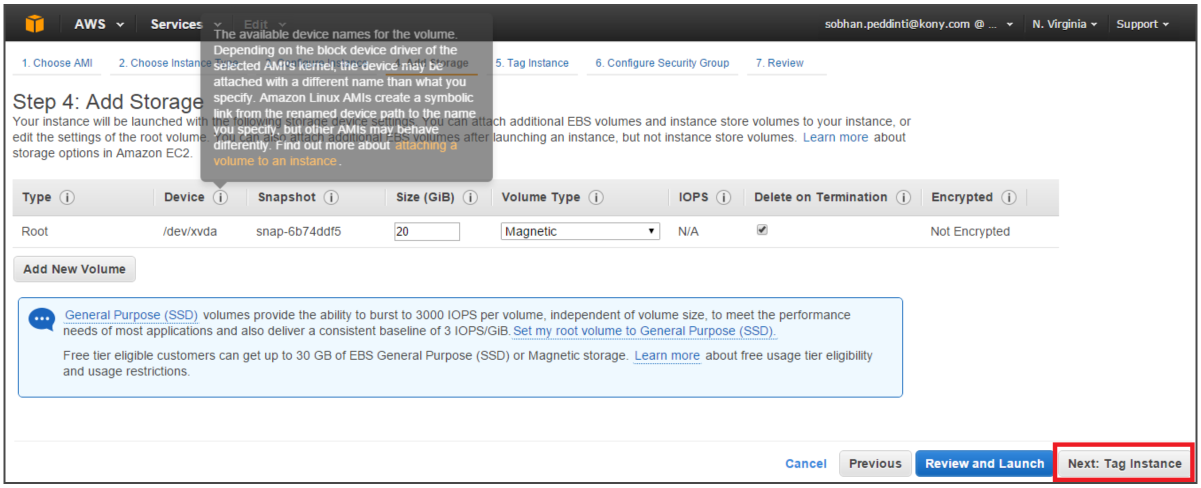
Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring

Tenancy: Shared tenancy (multi-tenant hardware)

9. In the **Step 4: Add Storage** page, leave the defaults settings and click the **Next: Tag Instance** button, shown below:



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

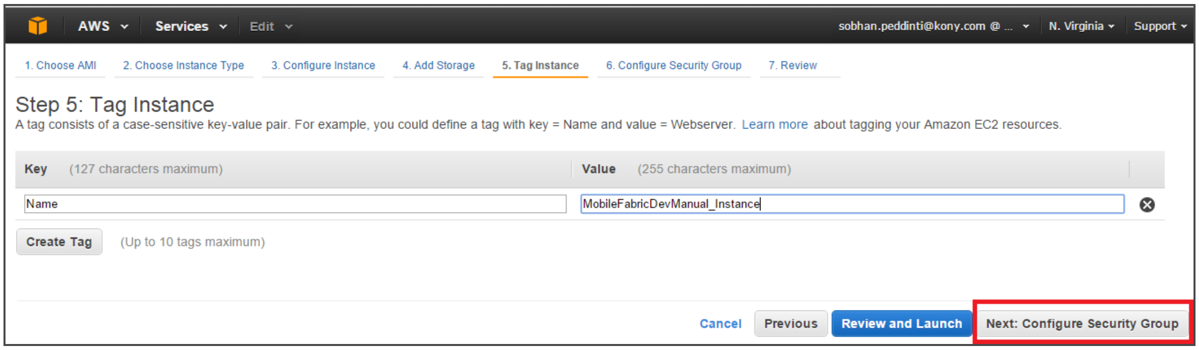
The available device names for the volume. Depending on the block device driver of the selected AMI's kernel, the device may be attached with a different name than what you specify. Amazon Linux AMIs create a symbolic link from the renamed device path to the name you specify, but other AMIs may behave differently. Find out more about attaching a volume to an instance.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-6b74ddf5	20	Magnetic	N/A	☑	Not Encrypted

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB. Set my root volume to General Purpose (SSD).

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

10. In the **Step 5: Tag Instance** page, enter the instance name and click the **Next: Configure Security Group** button, shown below:



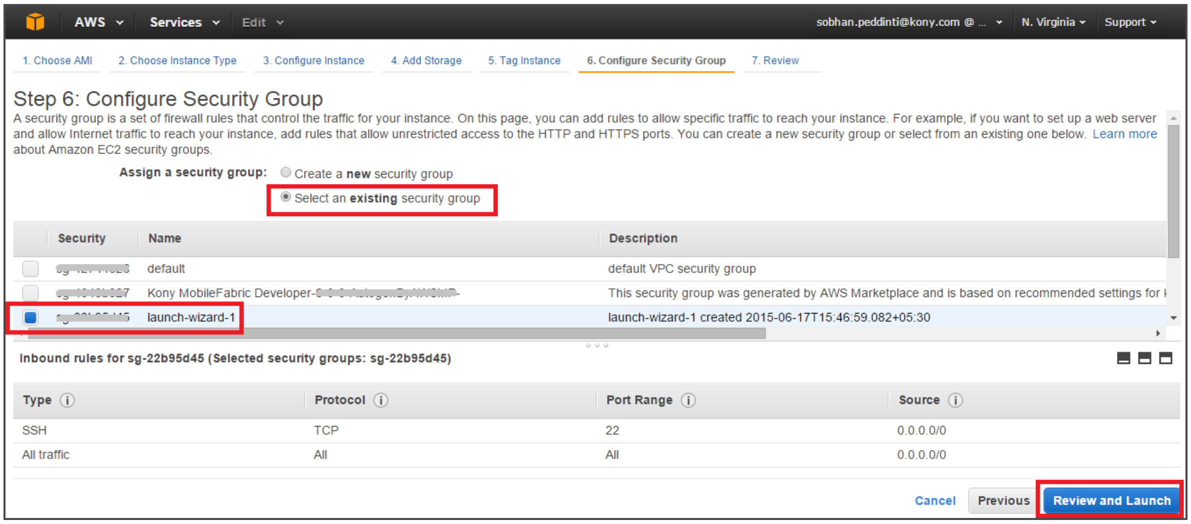
Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	MobileFabricDevManual_Instance

(Up to 10 tags maximum)

11. In the **Step 6: Configure Security Group** page, under the **Assign a security group**, click the **Select an existing security group**. Select the security option and click the **Review and Launch** button.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

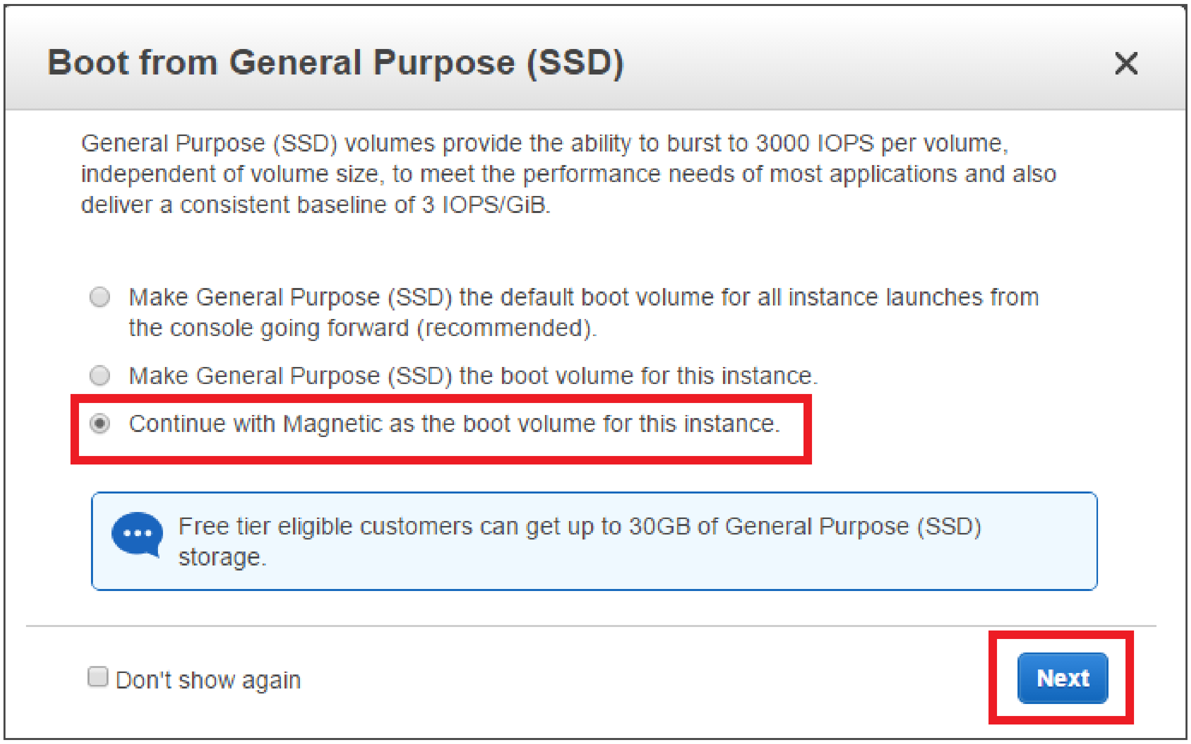
Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security	Name	Description
□	[redacted]	default	default VPC security group
□	[redacted]	Kony MobileFabric Developer-[redacted]-	This security group was generated by AWS Marketplace and is based on recommended settings for
■	[redacted]	launch-wizard-1	launch-wizard-1 created 2015-06-17T15:46:59.082+05:30

Inbound rules for sg-22b95d45 (Selected security groups: sg-22b95d45)

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
All traffic	All	All	0.0.0.0/0

12. In the **Boot from General Purpose (SSD)** pop-up, choose the **Continue with Magnetic as the book volume for this instance** and click **Next**, shown below:



Boot from General Purpose (SSD)

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB.

○ Make General Purpose (SSD) the default boot volume for all instance launches from the console going forward (recommended).

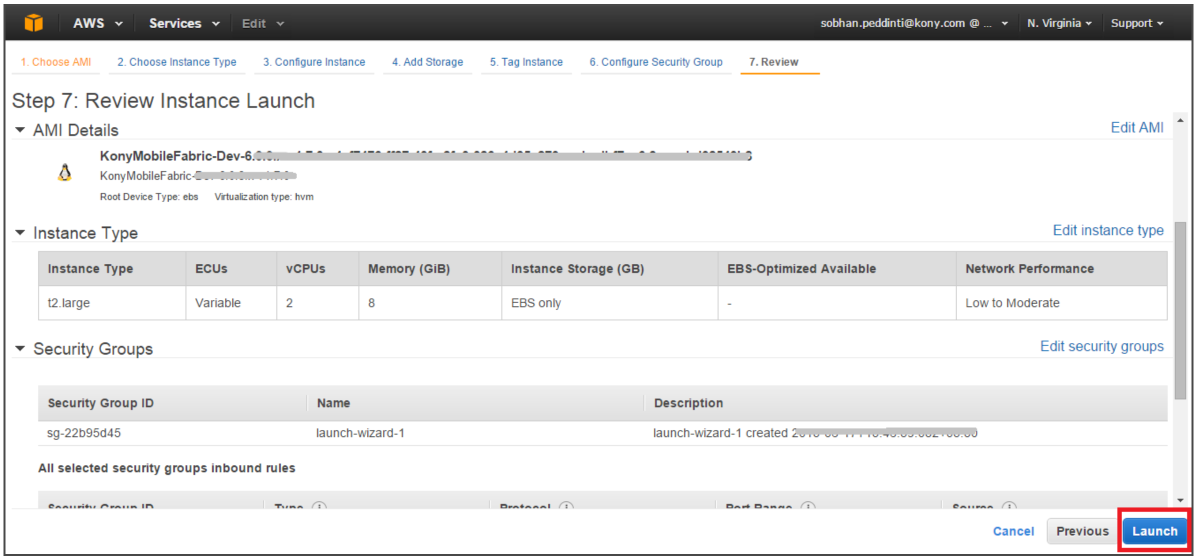
○ Make General Purpose (SSD) the boot volume for this instance.

● Continue with Magnetic as the boot volume for this instance.

Free tier eligible customers can get up to 30GB of General Purpose (SSD) storage.

□ Don't show again

13. In the **Step 7: Review Instance Launch** page, review the setting on the page and if correct, and click the **Launch** button.



Step 7: Review Instance Launch

AMI Details

KonyMobileFabric-Dev-6.[redacted]
KonyMobileFabric-[redacted]
Root Device Type: ebs   Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.large	Variable	2	8	EBS only	-	Low to Moderate

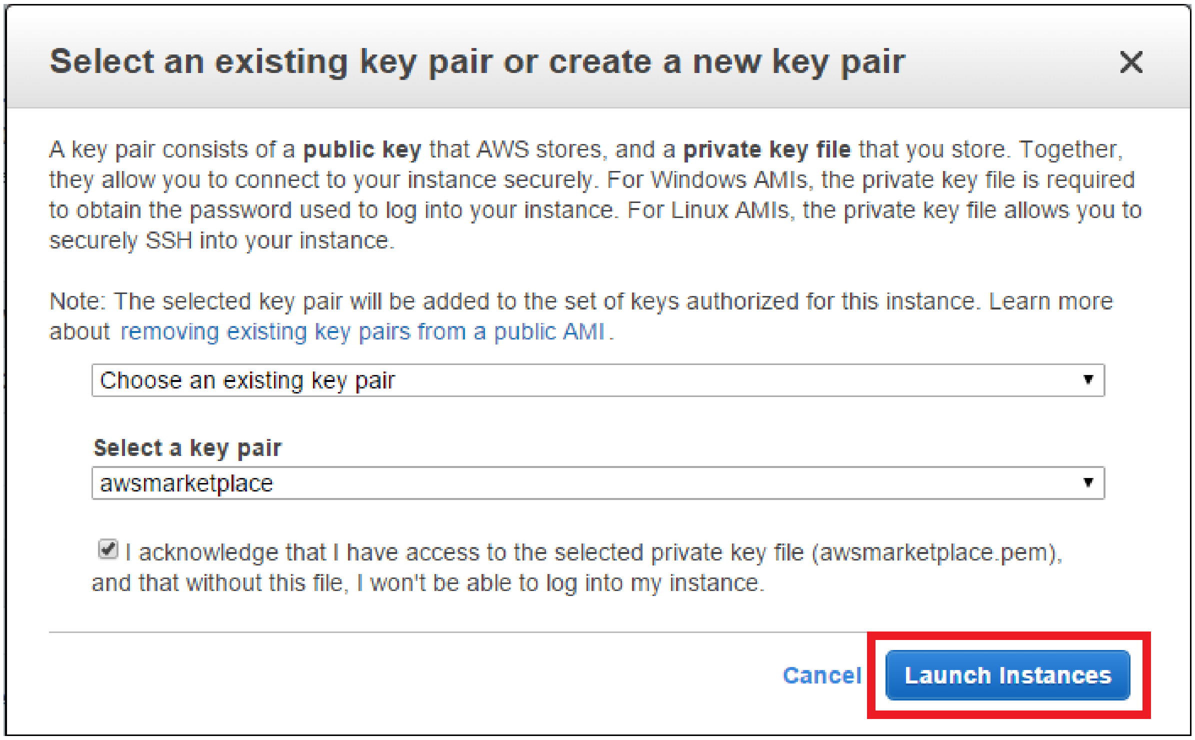
Security Groups

Security Group ID	Name	Description
sg-22b95d45	launch-wizard-1	launch-wizard-1 created 2[redacted]

All selected security groups inbound rules

Security Group ID	Type	Protocol	Port Range	Source

14. In the **Select an existing key pair or create a new key pair** pop-up, select the check box and click the **Launch Instances** button, shown below:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

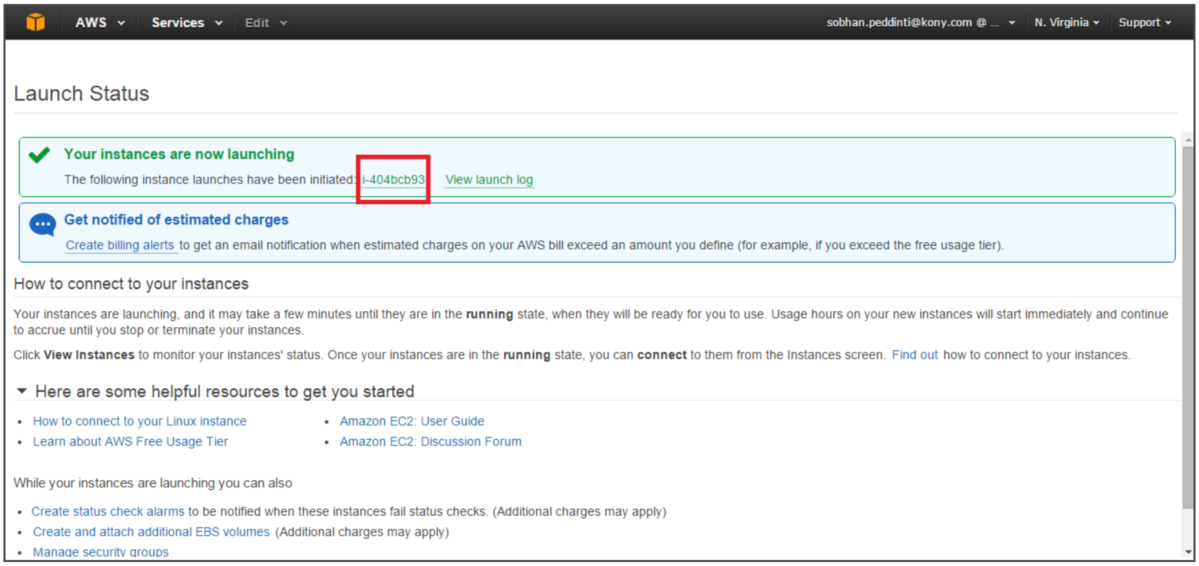
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair ▼

Select a key pair
awsmarketplace ▼

☑ I acknowledge that I have access to the selected private key file (awsmarketplace.pem), and that without this file, I won't be able to log into my instance.

15. In the **Launch Status** page that appears, click the link to manage your instances, shown below:



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-404bcb93 View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

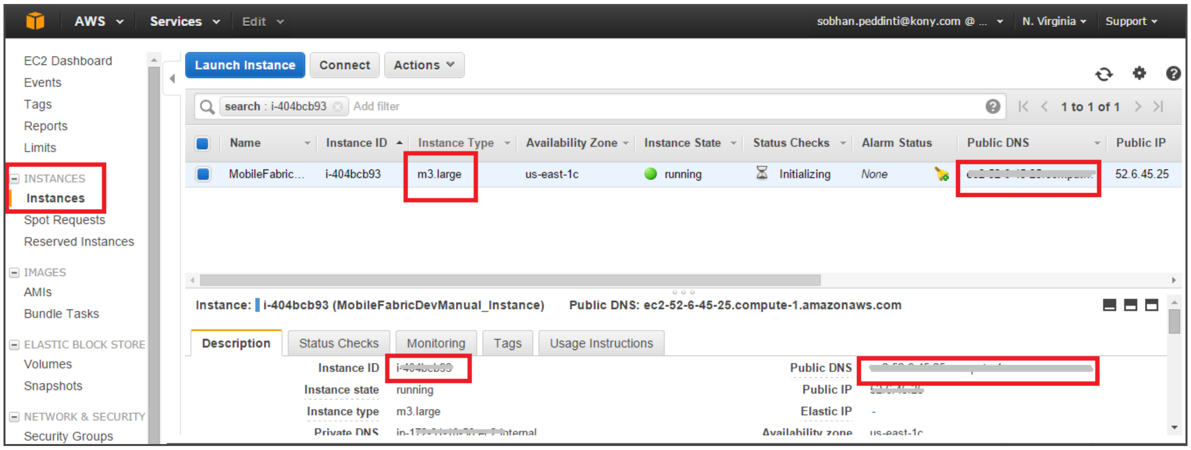
While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

16. The Instance details pop-up appears. Click the **Cancel** button to close the pop-up window.

In the **Instances**, the MobileFabric instance is created, shown below:

- **Instance Type** indicates the instance type that you have chosen.
- **Public DNS** indicates URL for the MobileFabric Console.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
MobileFabric...	i-404bcb93	m3.large	us-east-1c	● running	Initializing	None	[redacted]	52.6.45.25

Instance: i-404bcb93 (MobileFabricDevManual_Instance) Public DNS: ec2-52-6-45-25.compute-1.amazonaws.com

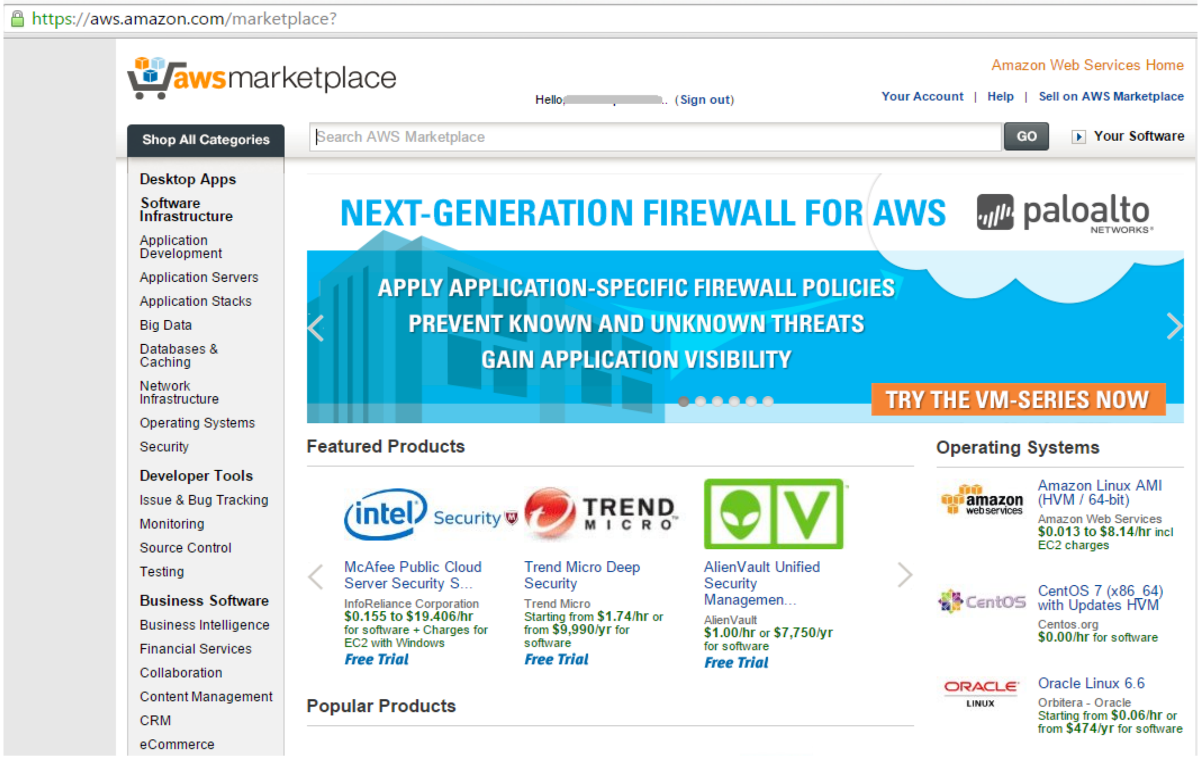
Instance ID	[redacted]	Public DNS	[redacted]
Instance state	running	Public IP	[redacted]
Instance type	m3.large	Elastic IP	-
Private DNS	ip-1[redacted]internal	Availability zone	us-east-1c

You can now launch the MobileFabric Instance.

2.3 Configuring Kony MobileFabric Express: 1-Click Launch

To 1-Click launch Kony MobileFabric Express Edition, follow these steps:

1. Go to https://aws.amazon.com/marketplace.
2. Sign in to your AWS Account.
3. In the AWS Market place screen, in the Search text box, type **Kony MobileFabric** and press **Enter**.

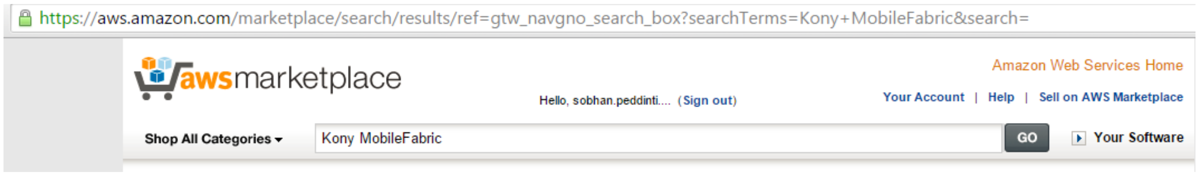


https://aws.amazon.com/marketplace?

Hello, [redacted] (Sign out)

Search AWS Marketplace

4. In the Search results, click Kony MobileFabric.

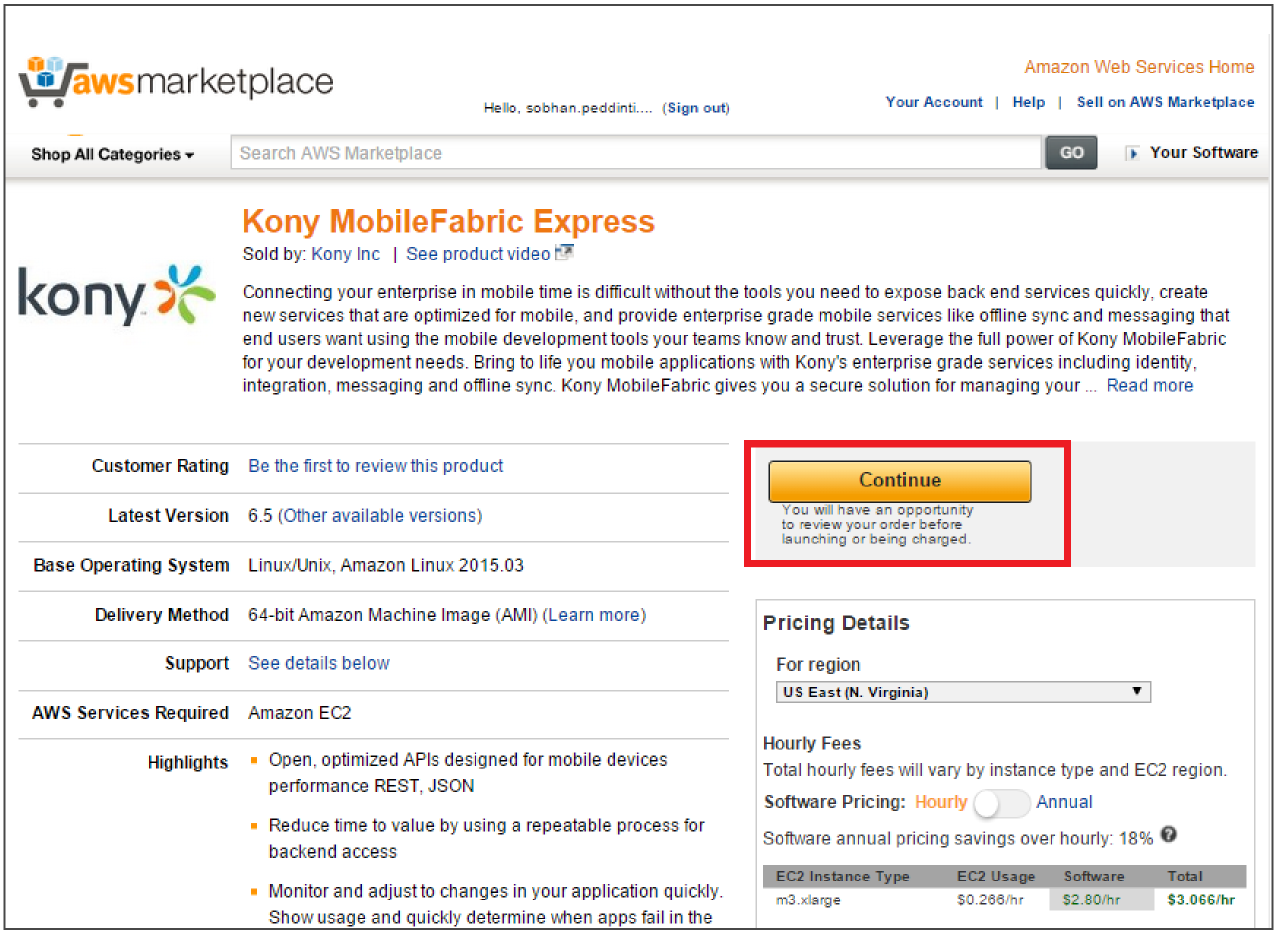


https://aws.amazon.com/marketplace/search/results/ref=gtw_navgno_search_box?searchTerms=Kony+MobileFabric&search=

Hello, sobhan.peddinti.... (Sign out)

Kony MobileFabric

5. **Kony MobileFabric Express** details page appears. Click **Continue**.



Kony MobileFabric Express

Sold by: Kony Inc | See product video

Connecting your enterprise in mobile time is difficult without the tools you need to expose back end services quickly, create new services that are optimized for mobile, and provide enterprise grade mobile services like offline sync and messaging that end users want using the mobile development tools your teams know and trust. Leverage the full power of Kony MobileFabric for your development needs. Bring to life you mobile applications with Kony's enterprise grade services including identity, integration, messaging and offline sync. Kony MobileFabric gives you a secure solution for managing your ... Read more

Customer Rating	Be the first to review this product
Latest Version	6.5 (Other available versions)
Base Operating System	Linux/Unix, Amazon Linux 2015.03
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	See details below
AWS Services Required	Amazon EC2
Highlights	▪ Open, optimized APIs designed for mobile devices performance REST, JSON ▪ Reduce time to value by using a repeatable process for backend access ▪ Monitor and adjust to changes in your application quickly. Show usage and quickly determine when apps fail in the

You will have an opportunity to review your order before launching or being charged.

Pricing Details

For region

US East (N. Virginia)

Hourly Fees

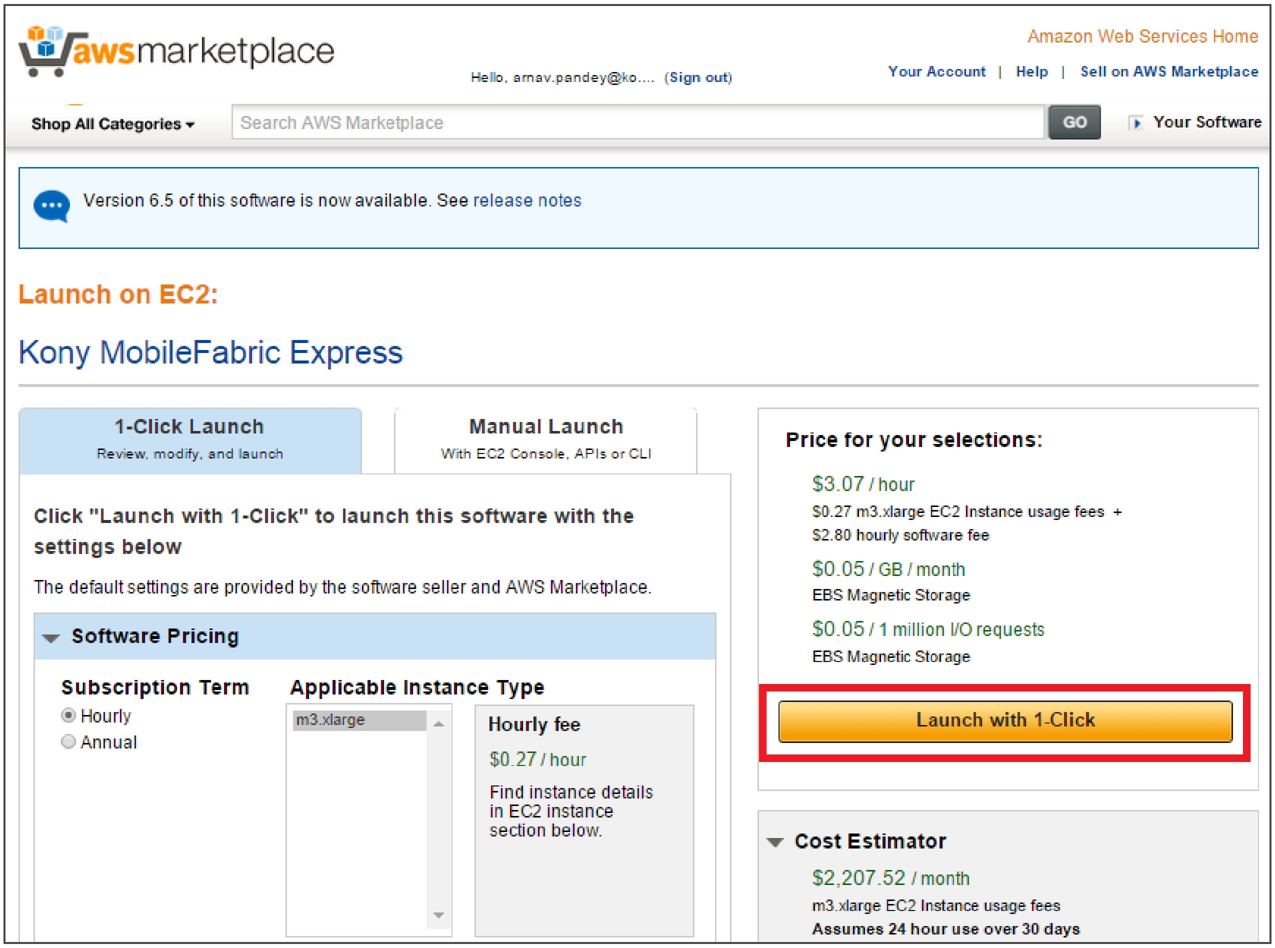
Total hourly fees will vary by instance type and EC2 region.

Software Pricing: Hourly Annual

Software annual pricing savings over hourly: 18%

EC2 Instance Type	EC2 Usage	Software	Total
m3.xlarge	$0.266/hr	$2.80/hr	$3.066/hr

6. The **Launch on EC2** page appears. Under the **Kony MobileFabric Express**, click the **1-Click Launch** tab:



Amazon Web Services Home

Hello, arnav.pandey@ko.... (Sign out)

Your Account | Help | Sell on AWS Marketplace

Shop All Categories

Search AWS Marketplace

Your Software

Version 6.5 of this software is now available. See release notes

Launch on EC2:

Kony MobileFabric Express

1-Click Launch
Review, modify, and launch

Manual Launch
With EC2 Console, APIs or CLI

Click "Launch with 1-Click" to launch this software with the settings below

The default settings are provided by the software seller and AWS Marketplace.

Software Pricing

Subscription Term
● Hourly
○ Annual

Applicable Instance Type
m3.xlarge

Hourly fee
$0.27 / hour
Find instance details in EC2 instance section below.

Price for your selections:

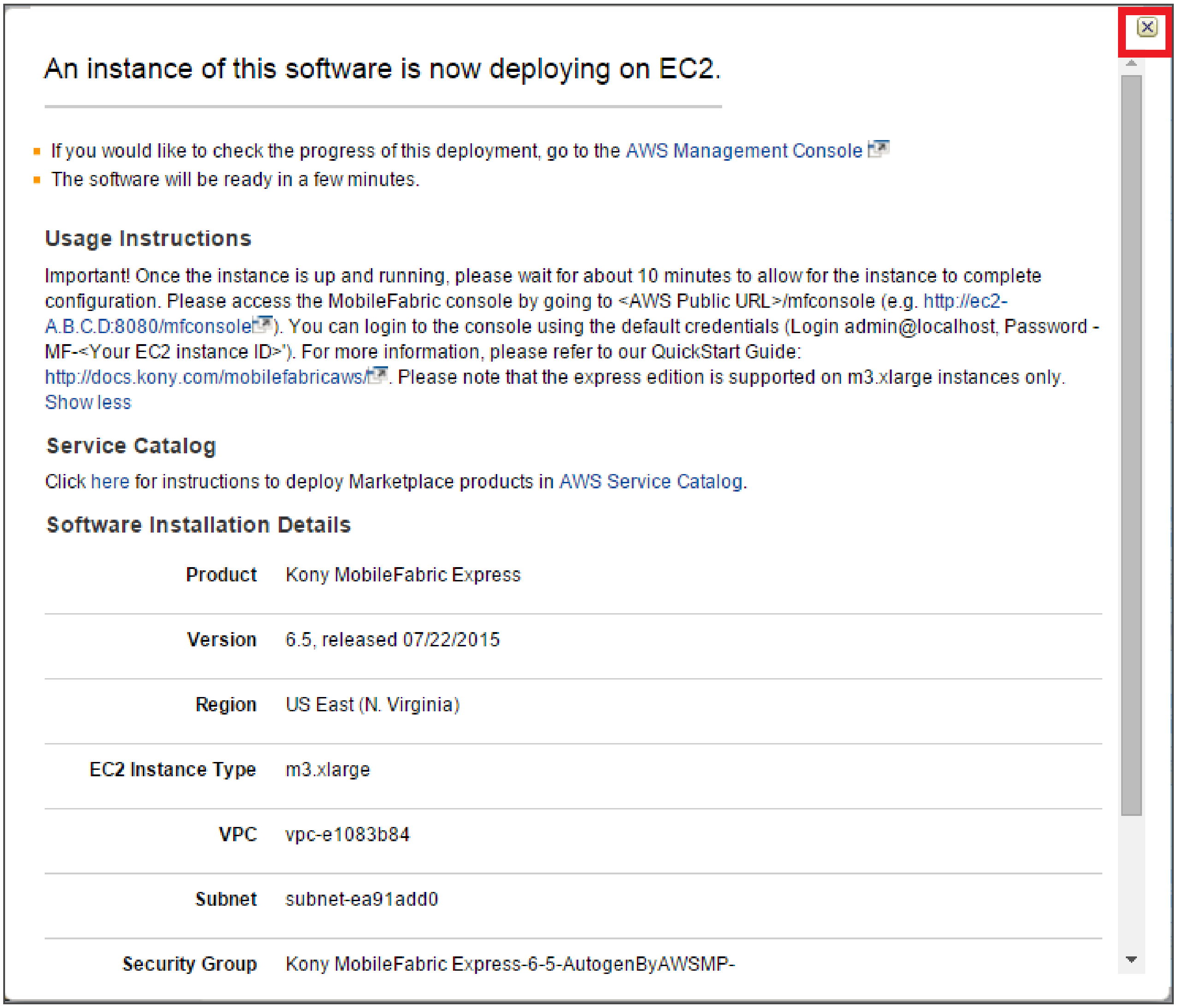
$3.07 / hour
$0.27 m3.xlarge EC2 Instance usage fees +
$2.80 hourly software fee

$0.05 / GB / month
EBS Magnetic Storage

$0.05 / 1 million I/O requests
EBS Magnetic Storage

Cost Estimator
$2,207.52 / month
m3.xlarge EC2 Instance usage fees
Assumes 24 hour use over 30 days

7. The Instance details pop-up appears. Click the **Cancel** button to close the pop-up window.



An instance of this software is now deploying on EC2.

- If you would like to check the progress of this deployment, go to the AWS Management Console
- The software will be ready in a few minutes.

Usage Instructions

Important! Once the instance is up and running, please wait for about 10 minutes to allow for the instance to complete configuration. Please access the MobileFabric console by going to <AWS Public URL>/mfconsole (e.g. http://ec2-A.B.C.D:8080/mfconsole). You can login to the console using the default credentials (Login admin@localhost, Password - MF-<Your EC2 instance ID>'). For more information, please refer to our QuickStart Guide: http://docs.kony.com/mobilefabricaws/. Please note that the express edition is supported on m3.xlarge instances only.
Show less

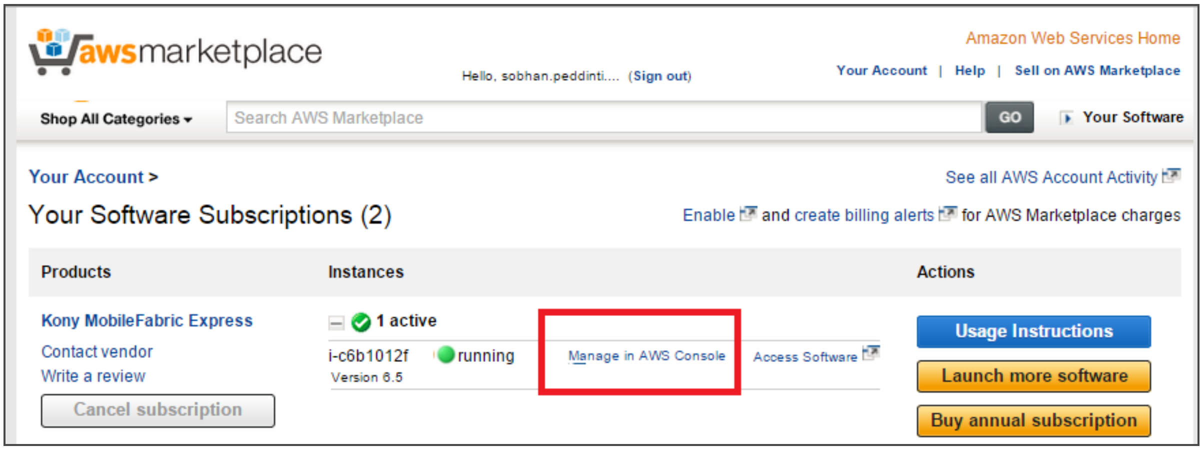
Service Catalog

Click here for instructions to deploy Marketplace products in AWS Service Catalog.

Software Installation Details

Product	Kony MobileFabric Express
Version	6.5, released 07/22/2015
Region	US East (N. Virginia)
EC2 Instance Type	m3.xlarge
VPC	vpc-e1083b84
Subnet	subnet-ea91add0
Security Group	Kony MobileFabric Express-6-5-AutogenByAWSMP-

8. In the page that appears, click the **Manage in AWS Console** link to manage your instances, shown below:

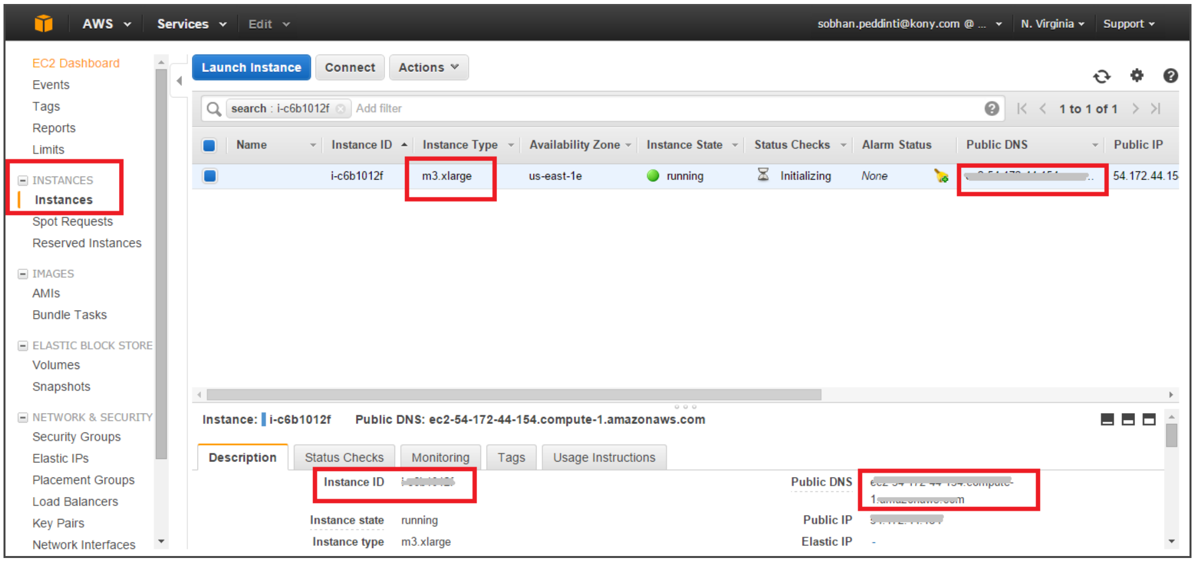


Your Software Subscriptions (2)

Products	Instances			Actions
Kony MobileFabric Express Contact vendor Write a review	1 active i-c6b1012f ● running Version 6.5	Manage in AWS Console	Access Software	Usage Instructions Launch more software Buy annual subscription

In the **Instances** page, the MobileFabric instance is created, shown below:

- **Instance Type** indicates the instance type that you have chosen.
- **Public DNS** indicates URL for the MobileFabric Console.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-c6b1012f	m3.xlarge	us-east-1e	running	Initializing	None	[redacted]	54.172.44.15

Instance: i-c6b1012f Public DNS: ec2-54-172-44-154.compute-1.amazonaws.com

Instance ID	[redacted]	Public DNS	[redacted]
Instance state	running	Public IP	[redacted]
Instance type	m3.xlarge	Elastic IP	-

You can now launch the MobileFabric Instance.

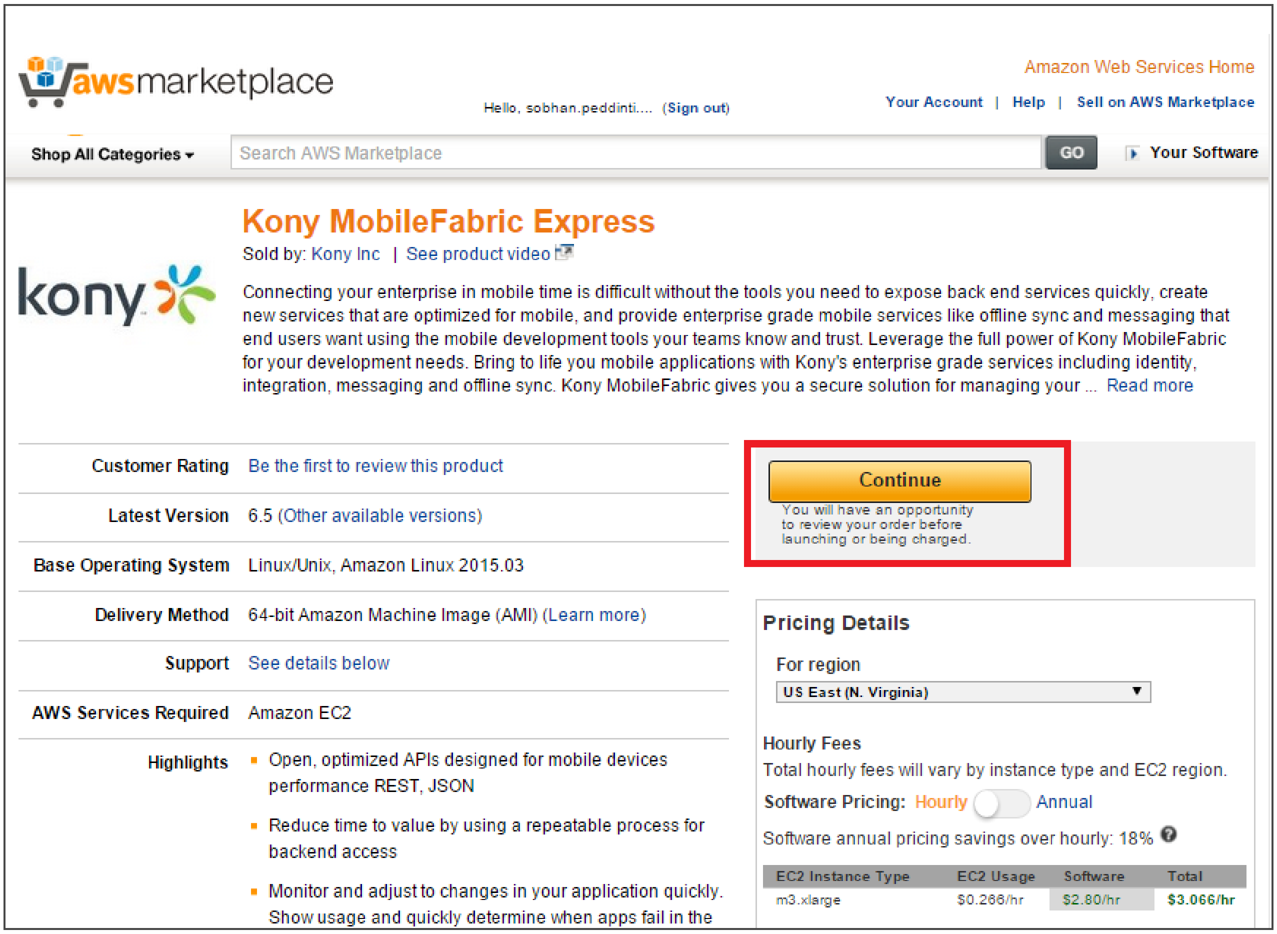
2.4 Configuring AWS Kony MobileFabric Express: Manual Launch

To manually launch Kony MobileFabric Express Edition, follow these steps:

1. Go to https://aws.amazon.com/marketplace.

2. Sign in to your AWS Account.

3. On the AWS Kony MobileFabric product page, click **Continue**.



Kony MobileFabric Express

Sold by: Kony Inc | See product video

Connecting your enterprise in mobile time is difficult without the tools you need to expose back end services quickly, create new services that are optimized for mobile, and provide enterprise grade mobile services like offline sync and messaging that end users want using the mobile development tools your teams know and trust. Leverage the full power of Kony MobileFabric for your development needs. Bring to life you mobile applications with Kony's enterprise grade services including identity, integration, messaging and offline sync. Kony MobileFabric gives you a secure solution for managing your ... Read more

Customer Rating	Be the first to review this product
Latest Version	6.5 (Other available versions)
Base Operating System	Linux/Unix, Amazon Linux 2015.03
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	See details below
AWS Services Required	Amazon EC2
Highlights	▪ Open, optimized APIs designed for mobile devices performance REST, JSON ▪ Reduce time to value by using a repeatable process for backend access ▪ Monitor and adjust to changes in your application quickly. Show usage and quickly determine when apps fail in the

You will have an opportunity to review your order before launching or being charged.

Pricing Details

For region: US East (N. Virginia)

Hourly Fees

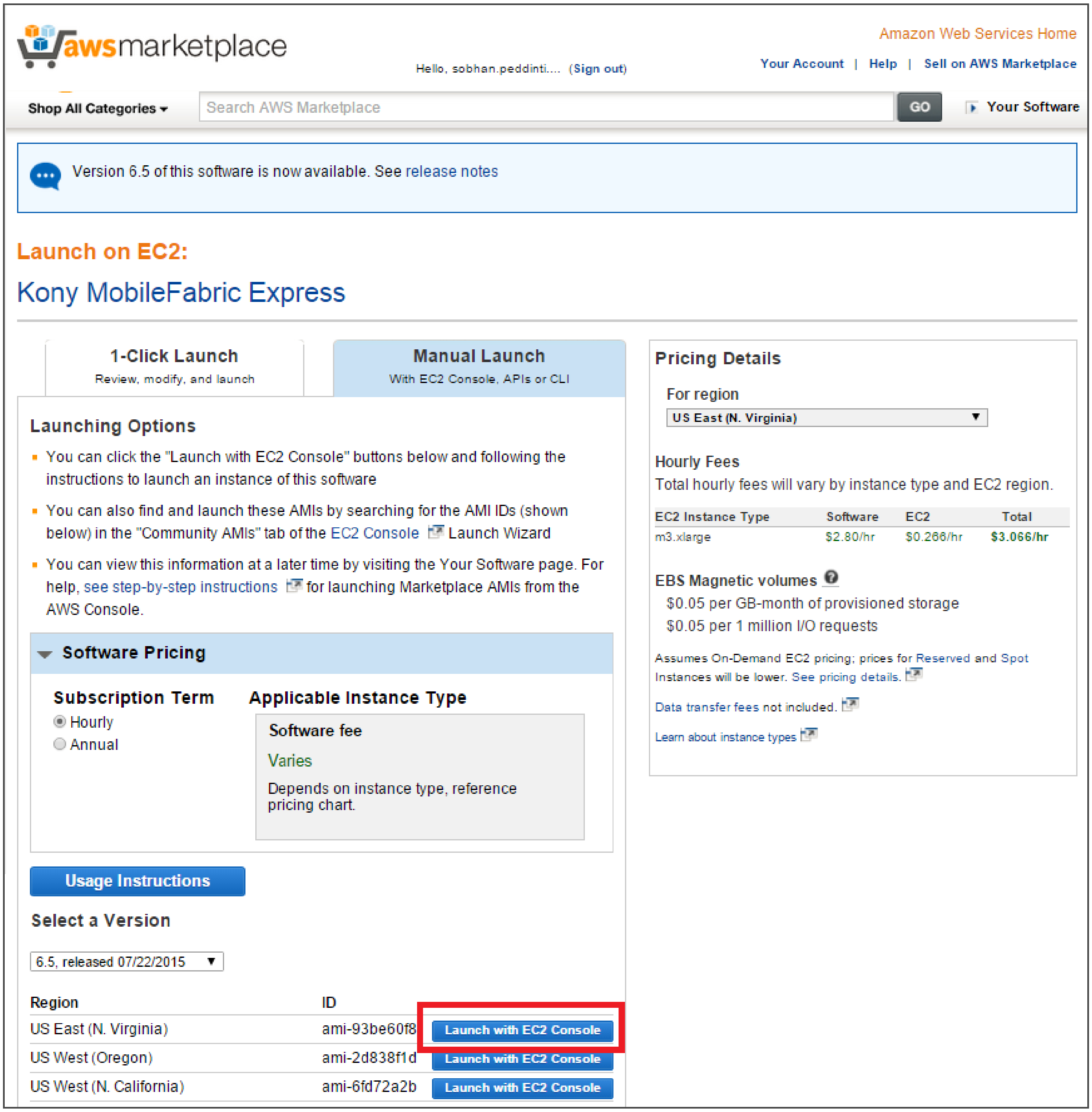
Total hourly fees will vary by instance type and EC2 region.

Software Pricing: Hourly ○ Annual

Software annual pricing savings over hourly: 18%

EC2 Instance Type	EC2 Usage	Software	Total
m3.xlarge	$0.266/hr	$2.80/hr	$3.066/hr

4. The **Launch on EC2** page appears. Under the **Kony MobileFabric Express**, click the **Manual Launch** tab.


awsmarketplace
Amazon Web Services Home
Hello, sobhan.peddinti.... (Sign out)
Your Account | Help | Sell on AWS Marketplace
Shop All Categories
Search AWS Marketplace
GO
Your Software

Version 6.5 of this software is now available. See release notes

Launch on EC2:

Kony MobileFabric Express

1-Click Launch
Review, modify, and launch
Manual Launch
With EC2 Console, APIs or CLI

Launching Options

- You can click the "Launch with EC2 Console" buttons below and following the instructions to launch an instance of this software
- You can also find and launch these AMIs by searching for the AMI IDs (shown below) in the "Community AMIs" tab of the EC2 Console Launch Wizard
- You can view this information at a later time by visiting the Your Software page. For help, see step-by-step instructions for launching Marketplace AMIs from the AWS Console.

Software Pricing

Subscription Term

● Hourly
○ Annual

Applicable Instance Type

Software fee

Varies

Depends on instance type, reference pricing chart.

Select a Version

6.5, released 07/22/2015

Region	ID	
US East (N. Virginia)	ami-93be60f8	Launch with EC2 Console
US West (Oregon)	ami-2d838f1d	Launch with EC2 Console
US West (N. California)	ami-6fd72a2b	Launch with EC2 Console

Pricing Details

For region

US East (N. Virginia)

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

EC2 Instance Type	Software	EC2	Total
m3.xlarge	$2.80/hr	$0.266/hr	$3.066/hr

EBS Magnetic volumes

$0.05 per GB-month of provisioned storage

$0.05 per 1 million I/O requests

Assumes On-Demand EC2 pricing; prices for Reserved and Spot Instances will be lower. See pricing details.

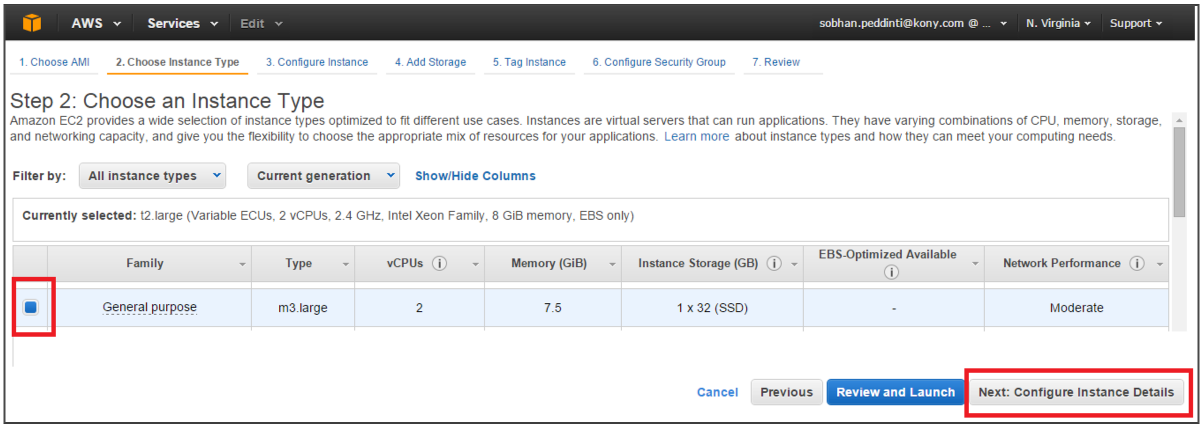
Data transfer fees not included.

Learn about instance types

5. In the **Manual Launch** tab, configure the required fields such as for region and version. In the **Region** section, click the **Launch with EC2 Console** button.

6. In the **Step 2: Choose an Instance Type** page, click the instance for type m3.large, shown below:

7. Click the **Next: Configure Instance Details** button.



Step 2: Choose an Instance Type

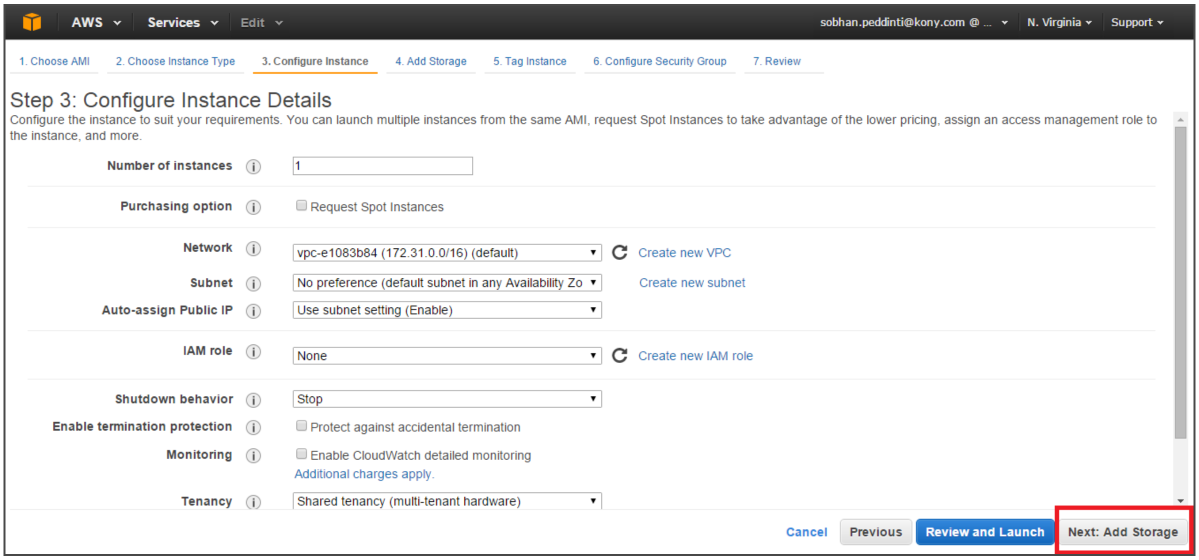
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.4 GHz, Intel Xeon Family, 8 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
■	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate

8. In the **Step 3: Configure Instance Details** page, leave the defaults settings and click the **Next: Add Storage** button, shown below:

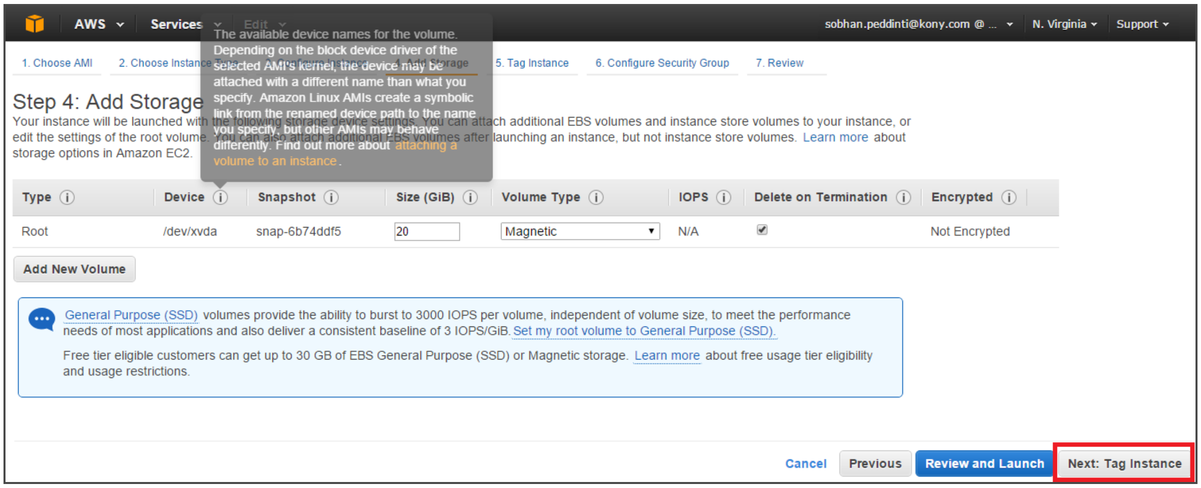


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	□ Request Spot Instances
Network	vpc-e1083b84 (172.31.0.0/16) (default)
Subnet	No preference (default subnet in any Availability Zo
Auto-assign Public IP	Use subnet setting (Enable)
IAM role	None
Shutdown behavior	Stop
Enable termination protection	□ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared tenancy (multi-tenant hardware)

9. In the **Step 4: Add Storage** page, leave the defaults settings and click the **Next: Tag Instance** button, shown below:



The available device names for the volume. Depending on the block device driver of the selected AMI's kernel, the device may be attached with a different name than what you specify. Amazon Linux AMIs create a symbolic link from the renamed device path to the name you specify, but other AMIs may behave differently. Find out more about attaching a volume to an instance.

Step 4: Add Storage

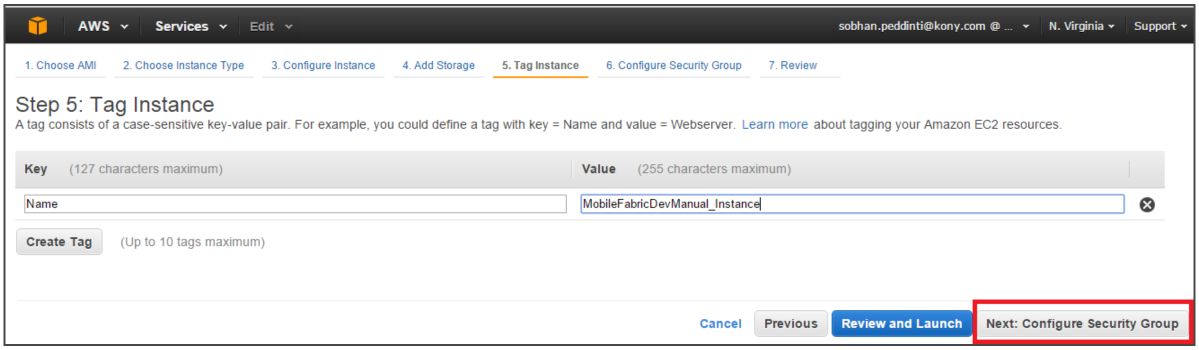
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-6b74ddf5	20	Magnetic	N/A	☑	Not Encrypted

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB. Set my root volume to General Purpose (SSD).

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

10. In the **Step 5: Tag Instance** page, enter the instance name and click the **Next: Configure Security Group** button, shown below:

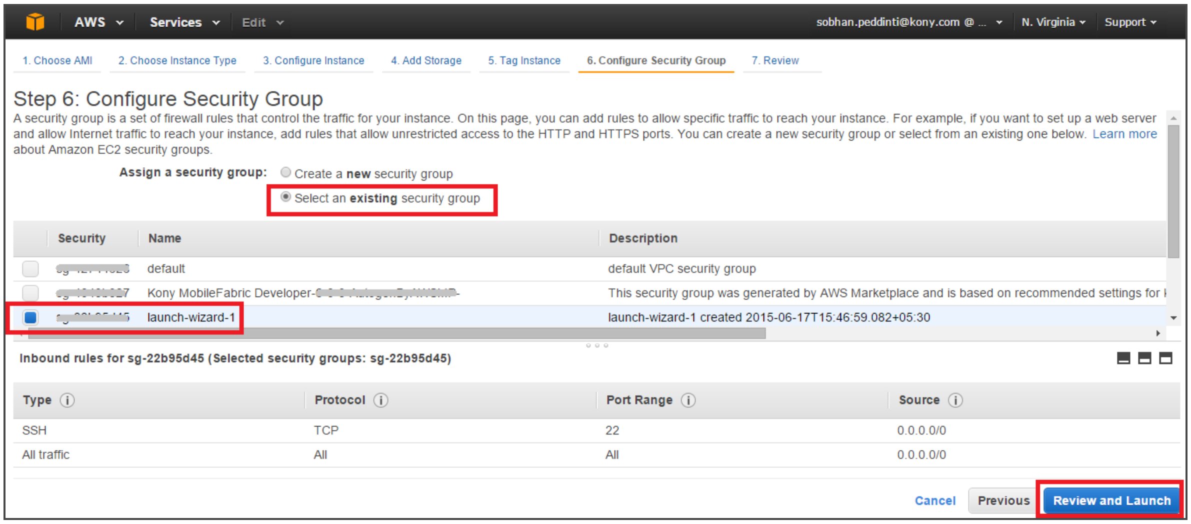


Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	MobileFabricDevManual_Instance

11. In the **Step 6: Configure Security Group** page, under the **Assign a security group**, click the **Select an existing security group**. Select the security option and click the **Review and Launch** button.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

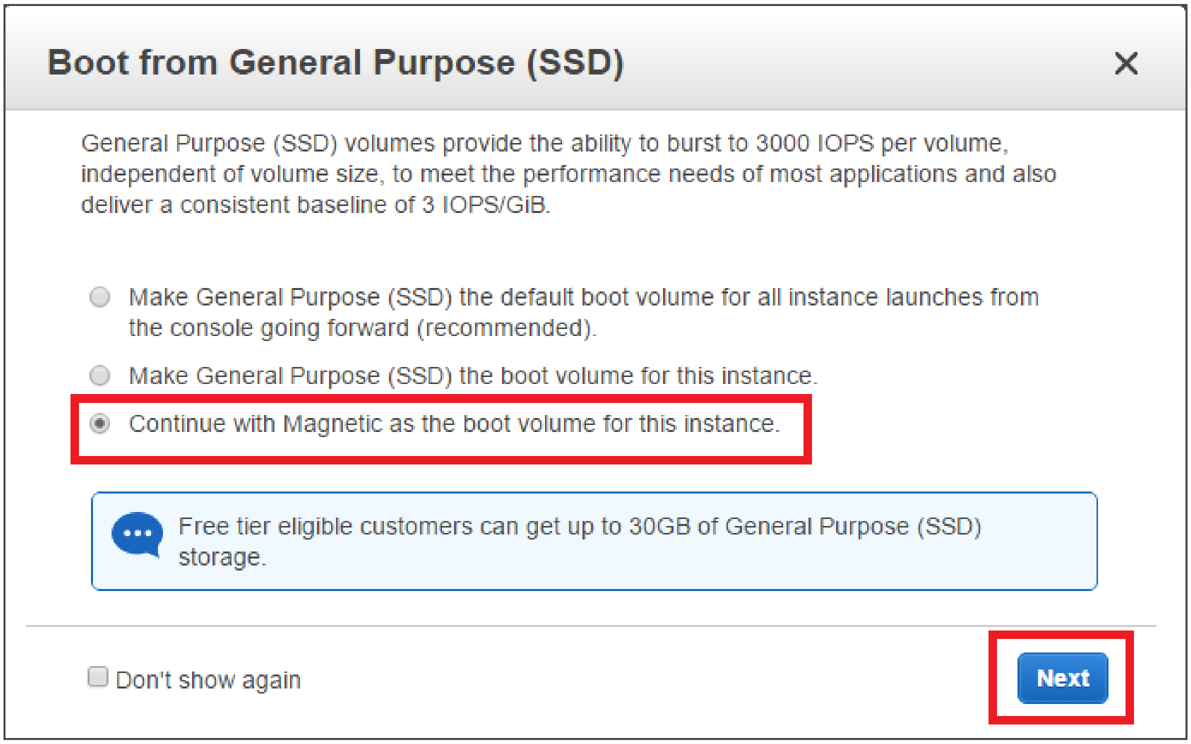
Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security	Name	Description
□	[redacted]	default	default VPC security group
□	[redacted]	Kony MobileFabric Developer-[redacted]-	This security group was generated by AWS Marketplace and is based on recommended settings for I
■	[redacted]	launch-wizard-1	launch-wizard-1 created 2015-06-17T15:46:59.082+05:30

Inbound rules for sg-22b95d45 (Selected security groups: sg-22b95d45)

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
All traffic	All	All	0.0.0.0/0

12. In the **Boot from General Purpose (SSD)** pop-up, choose the **Continue with Magnetic as the book volume for this instance** and click **Next**, shown below:



Boot from General Purpose (SSD)

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB.

○ Make General Purpose (SSD) the default boot volume for all instance launches from the console going forward (recommended).

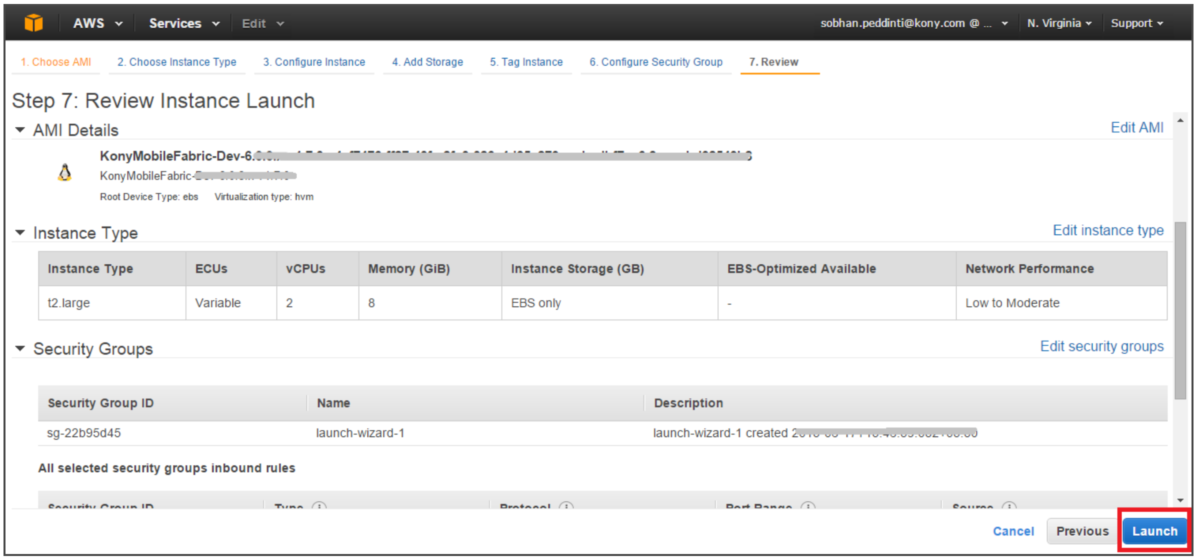
○ Make General Purpose (SSD) the boot volume for this instance.

● Continue with Magnetic as the boot volume for this instance.

Free tier eligible customers can get up to 30GB of General Purpose (SSD) storage.

□ Don't show again

13. In the **Step 7: Review Instance Launch** page, review the setting on the page and if correct, and click the **Launch** button.



Step 7: Review Instance Launch

AMI Details — Edit AMI

KonyMobileFabric-Dev-6.[redacted]
KonyMobileFabric-[redacted]
Root Device Type: ebs   Virtualization type: hvm

Instance Type — Edit instance type

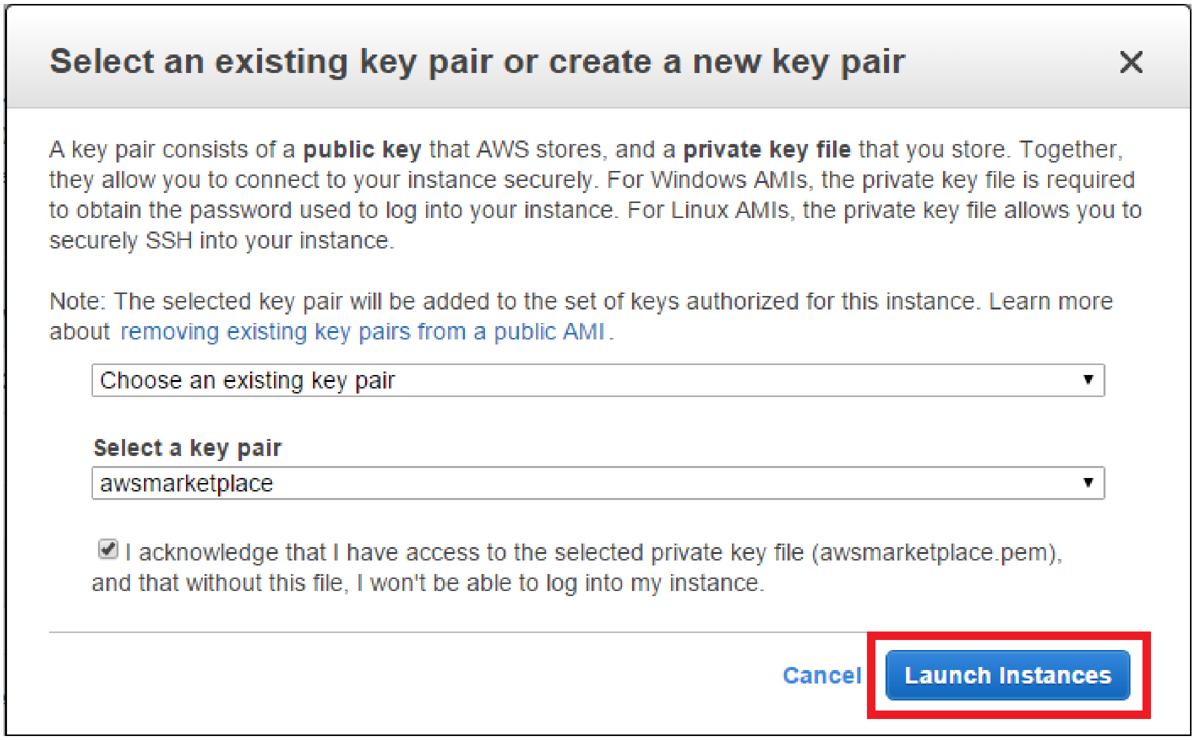
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.large	Variable	2	8	EBS only	-	Low to Moderate

Security Groups — Edit security groups

Security Group ID	Name	Description
sg-22b95d45	launch-wizard-1	launch-wizard-1 created 2[redacted]

All selected security groups inbound rules

14. In the **Select an existing key pair or create a new key pair** pop-up, select the check box and click the **Launch Instances** button, shown below:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

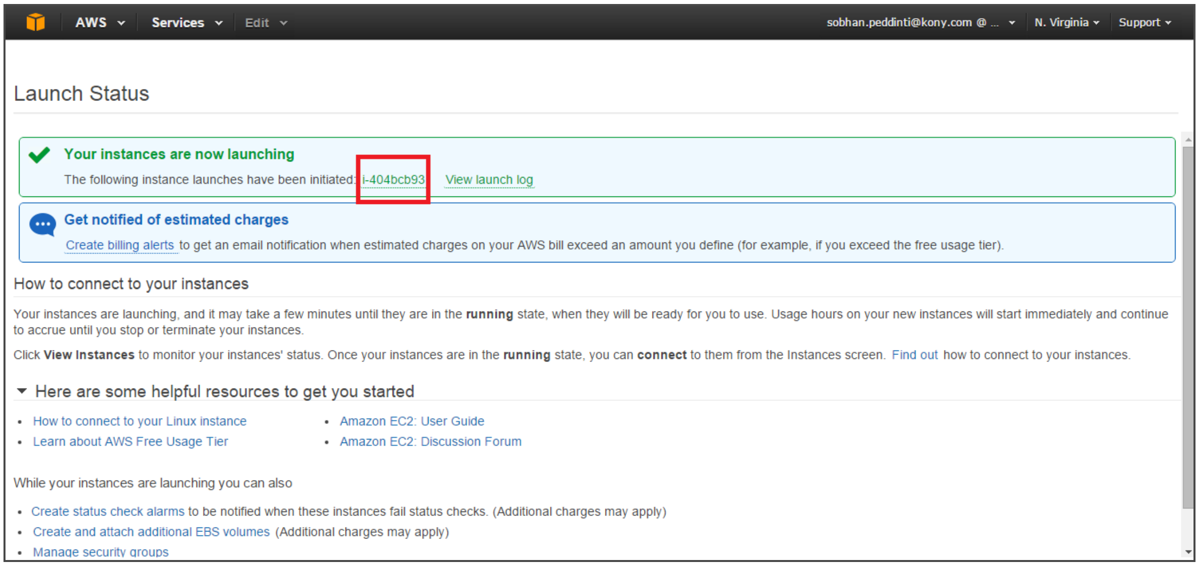
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair
awsmarketplace

☑ I acknowledge that I have access to the selected private key file (awsmarketplace.pem), and that without this file, I won't be able to log into my instance.

15. In the **Launch Status** page that appears, click the link to manage your instances, shown below:



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-404bcb93 View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

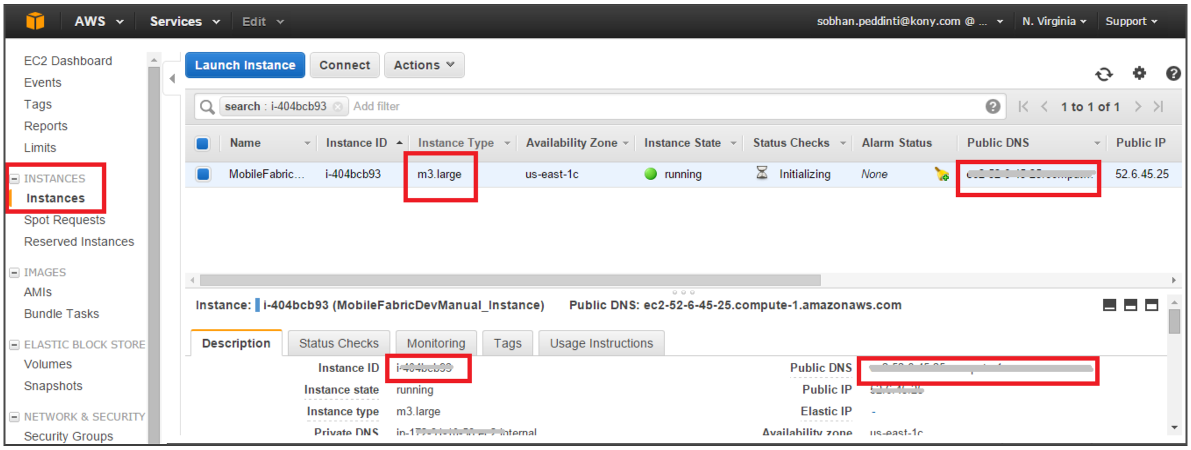
While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

16. The Instance details pop-up appears. Click the **Cancel** button to close the pop-up window.

In the **Instances**, the MobileFabric instance is created, shown below:

- **Instance Type** indicates the instance type that you have chosen.
- **Public DNS** indicates URL for the MobileFabric Console.



search : i-404bcb93

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
MobileFabric...	i-404bcb93	m3.large	us-east-1c	● running	Initializing	*None*	[redacted]	52.6.45.25

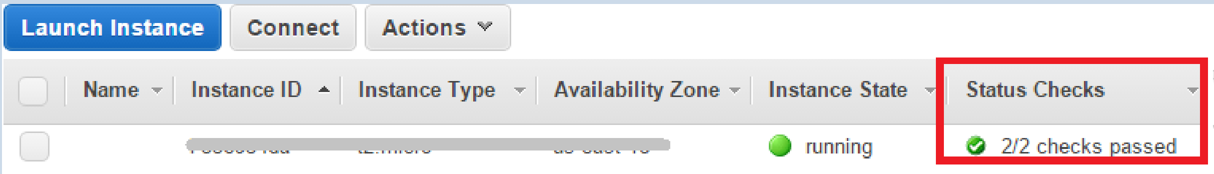
Instance: i-404bcb93 (MobileFabricDevManual_Instance)   Public DNS: ec2-52-6-45-25.compute-1.amazonaws.com

Instance ID	[redacted]	Public DNS	[redacted]
Instance state	running	Public IP	[redacted]
Instance type	m3.large	Elastic IP	-
Private DNS	ip-1[redacted]internal	Availability zone	us-east-1c

You can now launch the MobileFabric Instance.

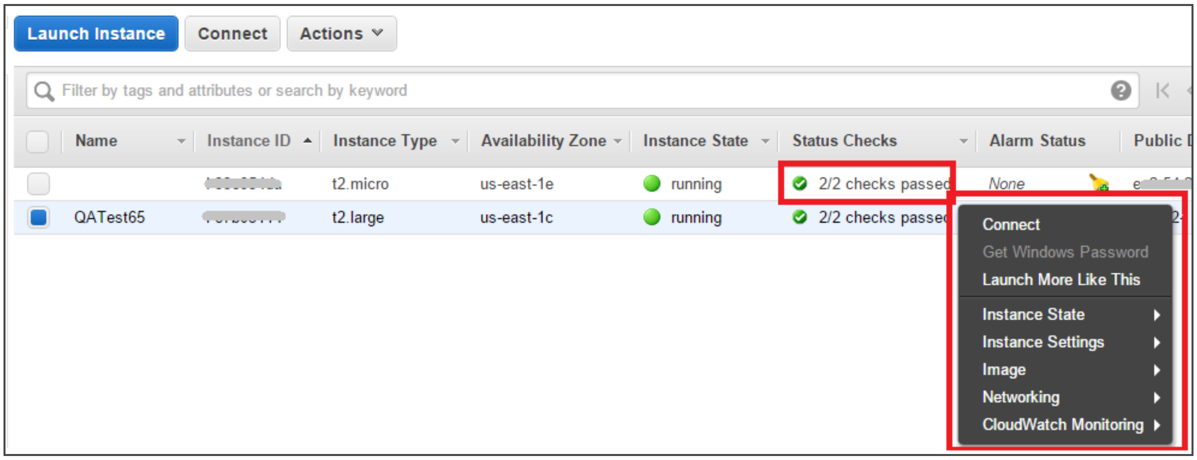
2.5 Launching the MobileFabric Instance

After your MobileFabric instance is created in the **Instances** section, follow these steps to launch the instance:

> *Important:* Before launching your instance, wait until the status changes to **checks passed** under the **Status Checks** column, shown below:
>
> 
>
>
> | □ | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks |
> |---|---|---|---|---|---|---|
> | □ | | [redacted] | [redacted] | [redacted] | ● running | ✓ 2/2 checks passed |

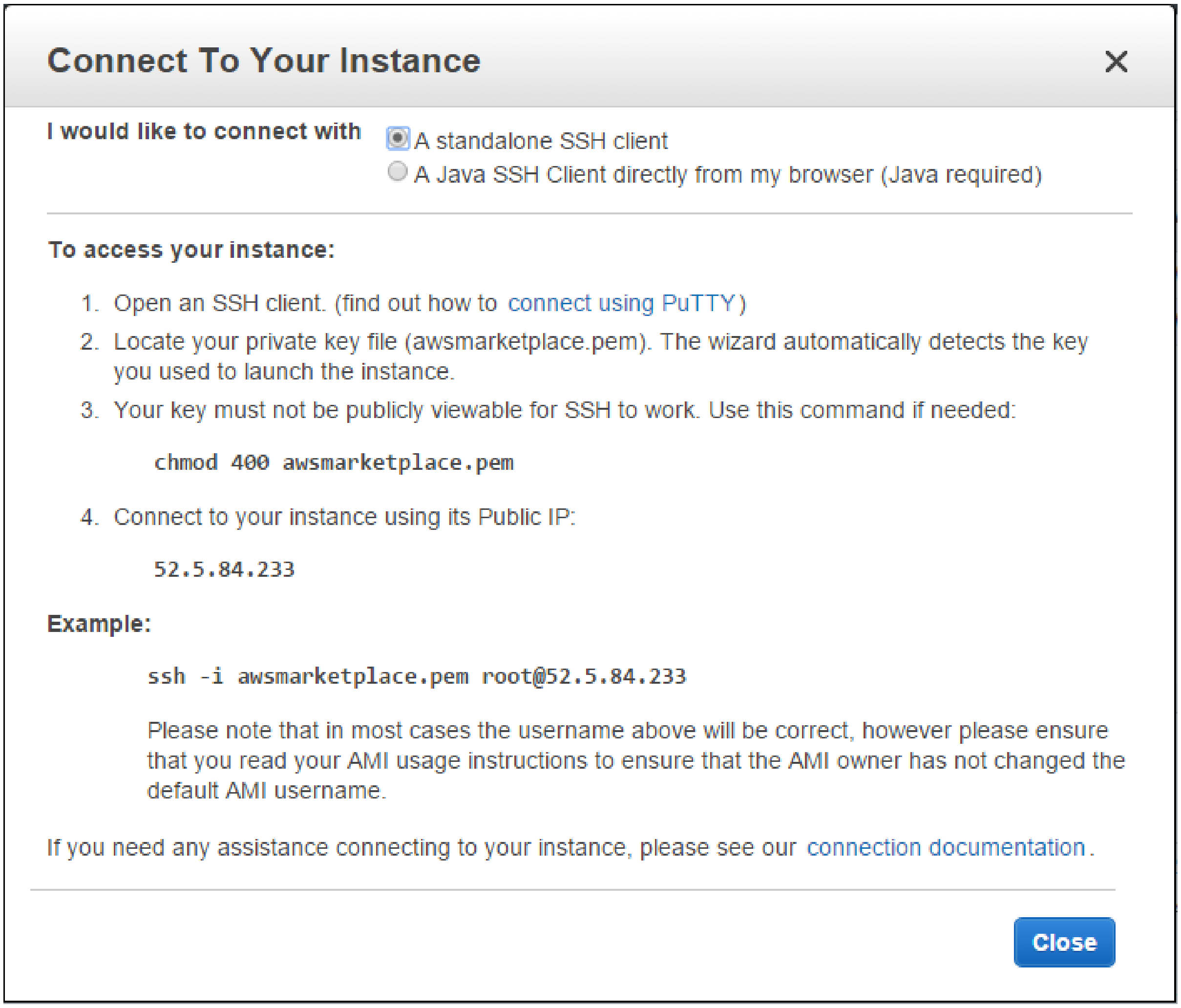
1. Assign a static IP to the selected instance.

2. In the **Instances** page, after your instance status is checks passed, right-click on the instance and choose **Connect**.



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public D
□		[redacted]	t2.micro	us-east-1e	● running	✓ 2/2 checks passed	None	e[redacted]
■	QATest65	[redacted]	t2.large	us-east-1c	● running	✓ 2/2 checks passe		

3. In the **Connect To Your Instance** dialog, follow the instructions to launch MobileFabric instance:



I would like to connect with
● A standalone SSH client
○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (awsmarketplace.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 awsmarketplace.pem
```

4. Connect to your instance using its Public IP:

```
52.5.84.233
```

Example:

```
ssh -i awsmarketplace.pem root@52.5.84.233
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

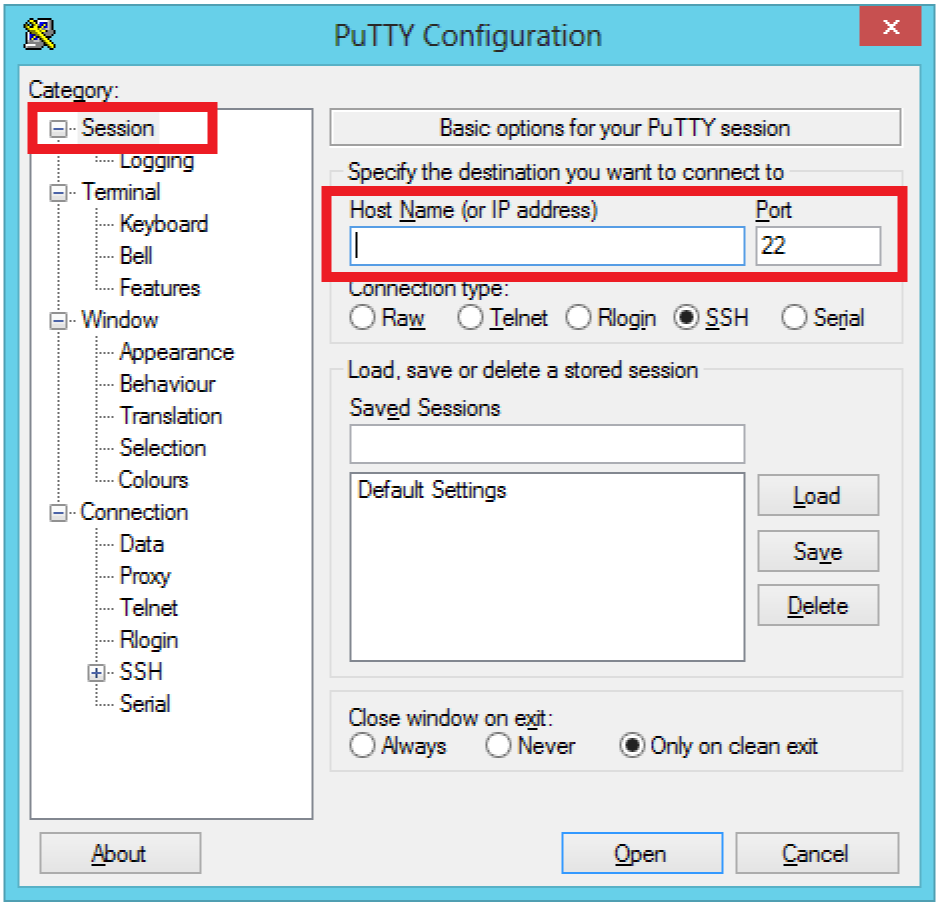
If you need any assistance connecting to your instance, please see our connection documentation.

For more details on connecting an instance, refer to AWS Documentation > How to connect instances.

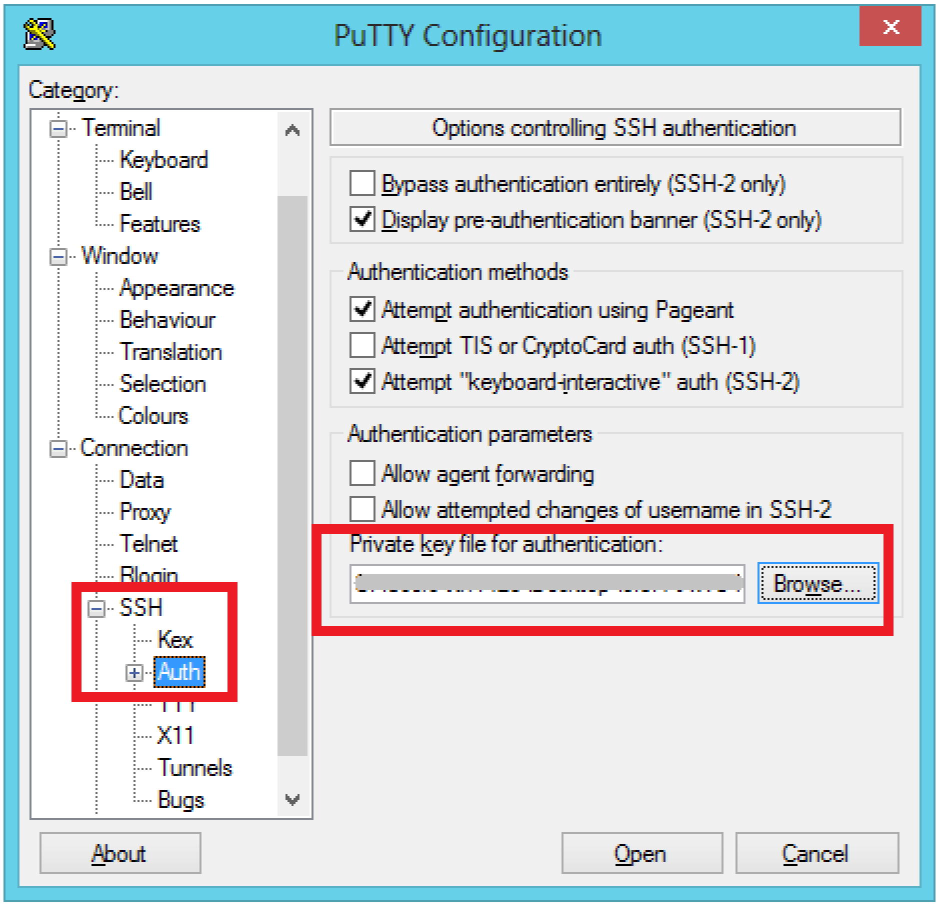
2.5.1 How to Launch MobileFabric instance using PuTTY

1. Open an **SSH** tool - for example, **PuTTy.exe** (www.putty.org/).
2. In the **Session > Host Name (or IP address)** text box, type the URL (Public DNS) that is generated for your instance.

Important: In the **Instances**, you can find the URL under the **Public DNS** column.



3. In **SSH > Auth**, click **Browse** to uploaded the Private key file for authentication - for example, a .ppk file.



Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

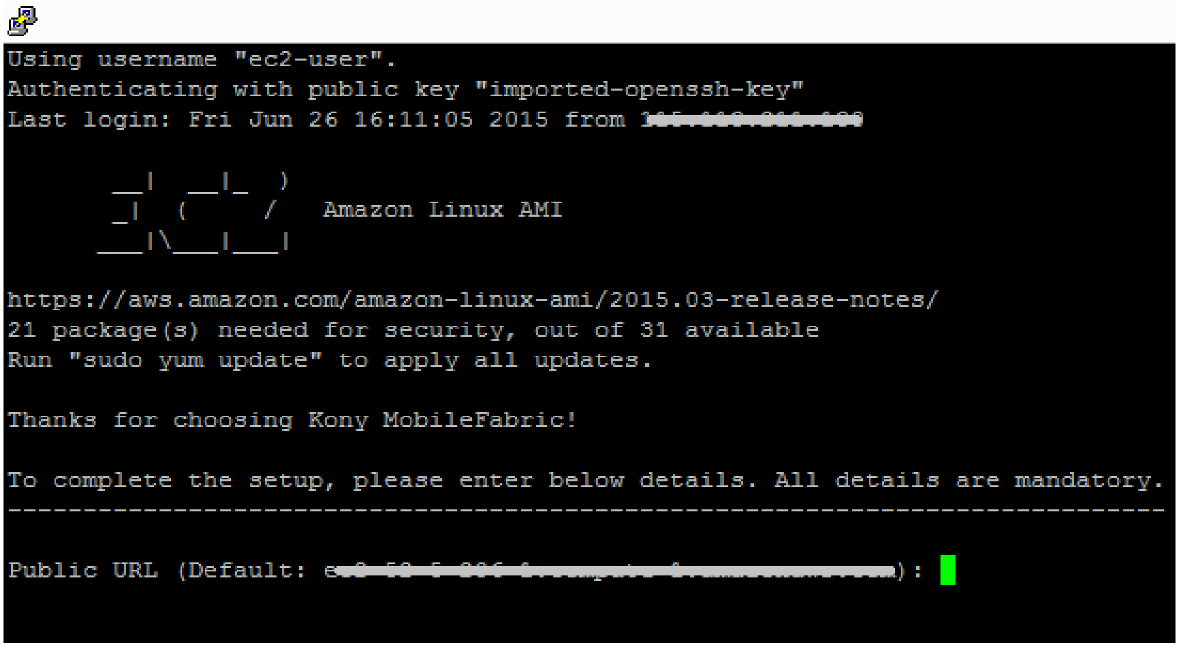
☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:
[redacted]

4. Click **Open** to start the setup process of MobileFabric components.

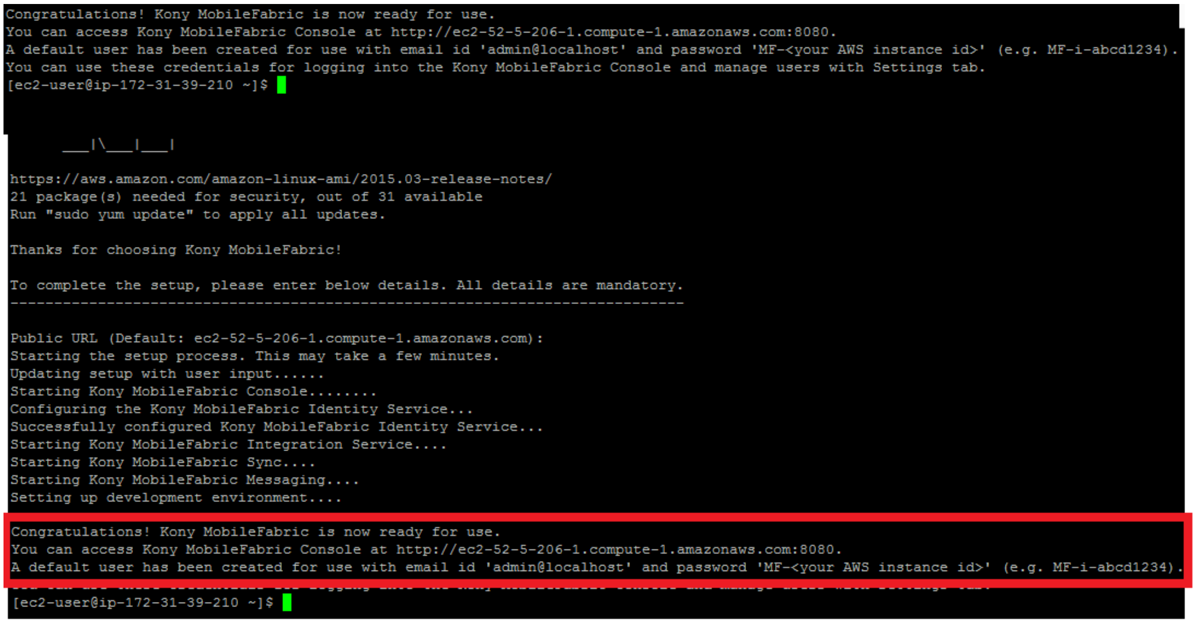


```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Fri Jun 26 16:11:05 2015 from [redacted]

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
21 package(s) needed for security, out of 31 available
Run "sudo yum update" to apply all updates.

Thanks for choosing Kony MobileFabric!

To complete the setup, please enter below details. All details are mandatory.
-----------------------------------------------------------------------------

Public URL (Default: [redacted]):
```

5. Provide valid DNS for the ec2 instance and press **Enter**. This setup may take a few minutes.

   After you run MobileFabric on AWS Marketplace, AWS validates the user credentials and installs the MobileFabric instance. During this stage, the system completes registration with Kony MobileFabric products such as Kony MobileFabric Identity and Console, Kony MobileFabric Integration, Kony MobileFabric Messaging, and Kony MobileFabric Sync. The system creates an environment in the MobileFabric Console.

   Once the MobileFabric installation is completed, you will see the message: `Congratulations! Kony MobileFabric is now ready for use` in the setup complete screen, shown below:



```
Congratulations! Kony MobileFabric is now ready for use.
You can access Kony MobileFabric Console at http://ec2-52-5-206-1.compute-1.amazonaws.com:8080.
A default user has been created for use with email id 'admin@localhost' and password 'MF-<your AWS instance id>' (e.g. MF-i-abcd1234).
You can use these credentials for logging into the Kony MobileFabric Console and manage users with Settings tab.
[ec2-user@ip-172-31-39-210 ~]$


      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
21 package(s) needed for security, out of 31 available
Run "sudo yum update" to apply all updates.

Thanks for choosing Kony MobileFabric!

To complete the setup, please enter below details. All details are mandatory.
-----------------------------------------------------------------------------

Public URL (Default: ec2-52-5-206-1.compute-1.amazonaws.com):
Starting the setup process. This may take a few minutes.
Updating setup with user input......
Starting Kony MobileFabric Console........
Configuring the Kony MobileFabric Identity Service...
Successfully configured Kony MobileFabric Identity Service...
Starting Kony MobileFabric Integration Service....
Starting Kony MobileFabric Sync....
Starting Kony MobileFabric Messaging....
Setting up development environment....

Congratulations! Kony MobileFabric is now ready for use.
You can access Kony MobileFabric Console at http://ec2-52-5-206-1.compute-1.amazonaws.com:8080.
A default user has been created for use with email id 'admin@localhost' and password 'MF-<your AWS instance id>' (e.g. MF-i-abcd1234).
You can use these credentials for logging into the Kony MobileFabric Console and manage users with Settings tab.
[ec2-user@ip-172-31-39-210 ~]$
```

The setup complete screen contains the following details that you require to log in MobileFabric Console:

- Kony MobileFabric Console URL: `http://<ec2-host>:8080`
- Default user login email ID: `admin@localhost`
- Default user login password: `MF-<your AWS instance id>`. For example, `MF-i-abcd1234`

> ***Note:*** The following are the individual console URLs for reference:
>
> - Kony MobileFabric Messaging URL: `http://<ec2-host>:8280/kpns`
> - Kony MobileFabric Sync URL: `http://<ec2-host>:8180/syncconsole`
> - Middleware URL: `http://<ec2-host>:8080/admin`

Note: To access analytics and reports, connect to the database to allow access to JasperReports Server. For reporting to work JasperReports Server should be able to connect to the MySQL database residing on the EC2 instance. For this you need to connect via SSH to the instance and grant permissions to the root user to connect to this database from JasperReports Server node.

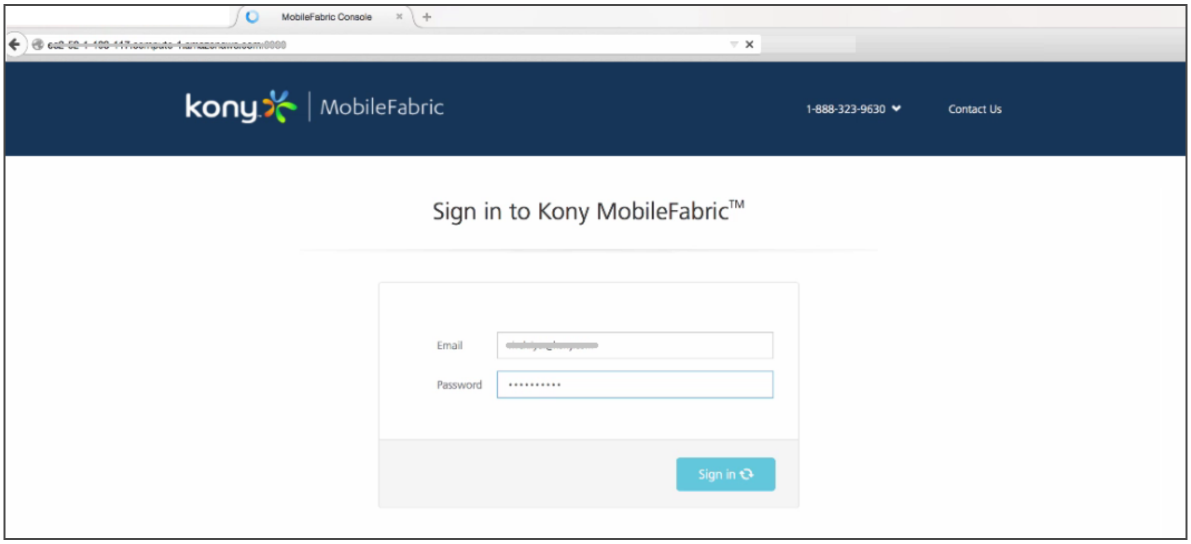
Use the following credentials to connect to the MySQL database:

- **User**: `root`
- **Password**: `root`

By default MySQL is blocked from connecting remotely. If required, you can connect to the MySQL DB on the same machine.

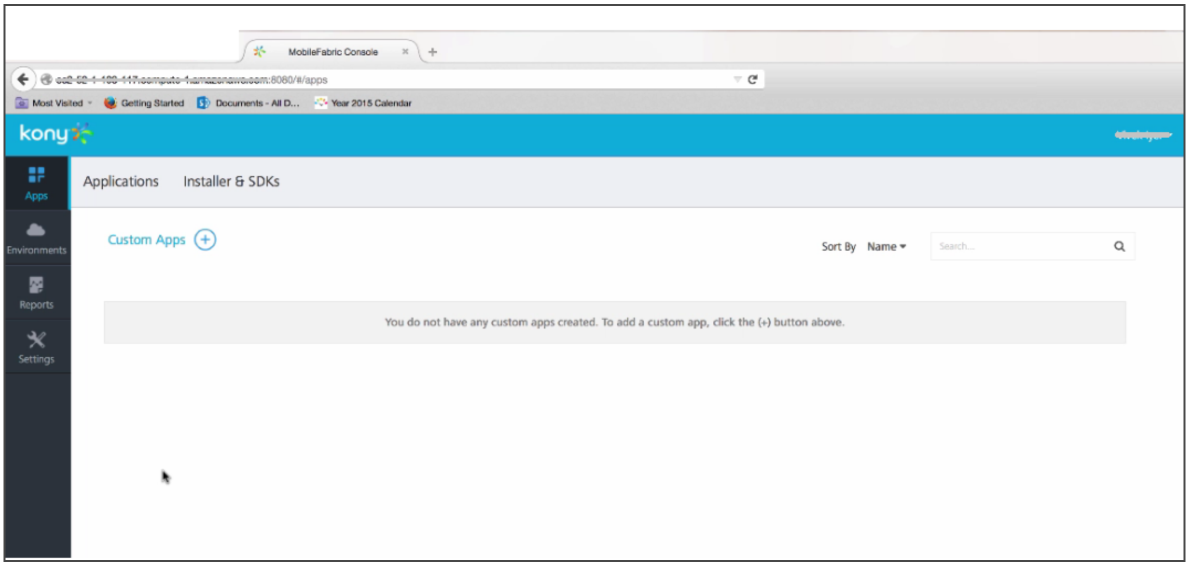
6. From the previous step, copy the `Kony MobileFabric Console URL` and paste it in a web browser's address bar and press the **Enter** key.

   The **Sign in to Kony MobileFabric** page appears.



7. Provide your instance account log-in credentials that are generated in the setup complete page, and click **Sign in**.

   After your credentials are authenticated, you are directed to your MobileFabric account. By default, the **Apps** page appears.

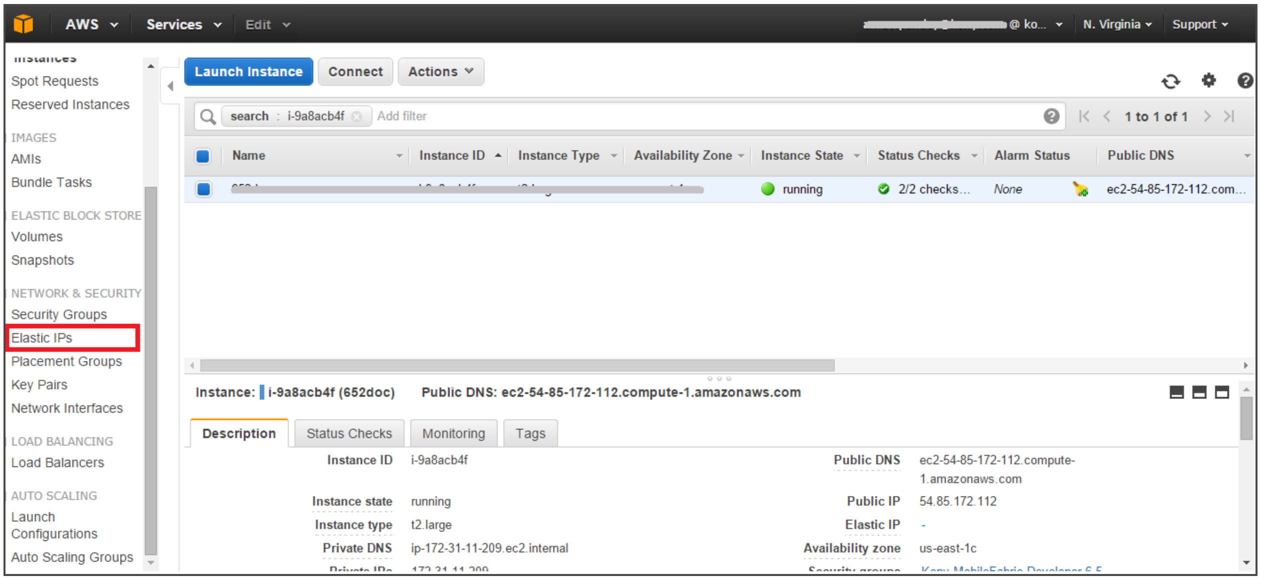


You do not have any custom apps created. To add a custom app, click the (+) button above.

2.6 Assigning Static IP for EC2 Instance

To assign a static IP for EC2 instance, follow these steps:

1. Log in to your AWS console and click **Elastic IPs**, shown below.



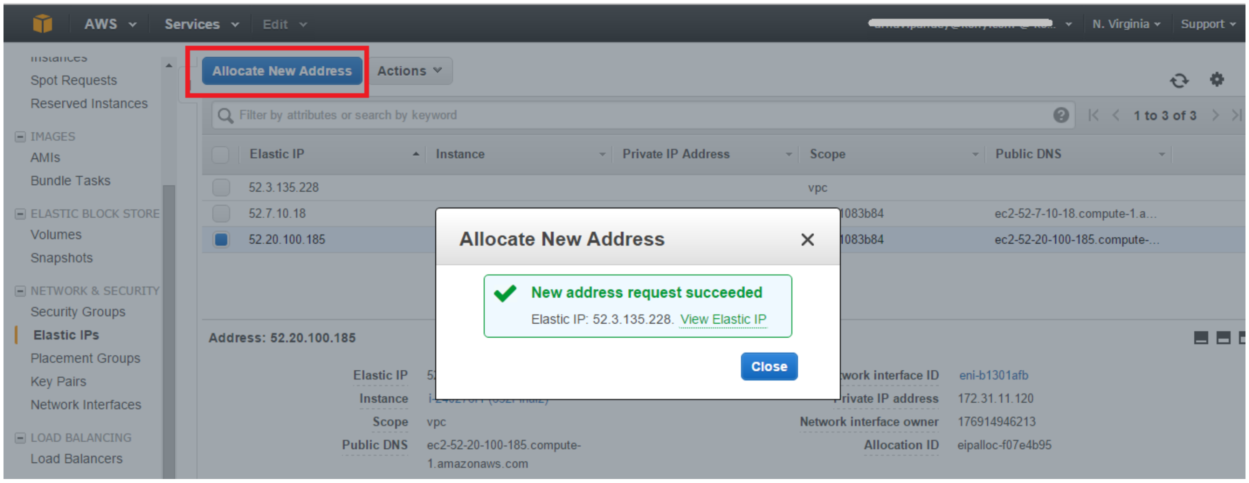
search : i-9a8acb4f

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
[redacted]				running	2/2 checks...	None	ec2-54-85-172-112.com...

Instance: i-9a8acb4f (652doc) Public DNS: ec2-54-85-172-112.compute-1.amazonaws.com

Instance ID	i-9a8acb4f	Public DNS	ec2-54-85-172-112.compute-1.amazonaws.com
Instance state	running	Public IP	54.85.172.112
Instance type	t2.large	Elastic IP	-
Private DNS	ip-172-31-11-209.ec2.internal	Availability zone	us-east-1c
Private IPs	172.31.11.209	Security groups	Kony MobileFabric Developer 6.5

2. Click **Allocate New Address**.



Elastic IP	Instance	Private IP Address	Scope	Public DNS
52.3.135.228			vpc	
52.7.10.18			1083b84	ec2-52-7-10-18.compute-1.a...
52.20.100.185			1083b84	ec2-52-20-100-185.compute-...

Allocate New Address

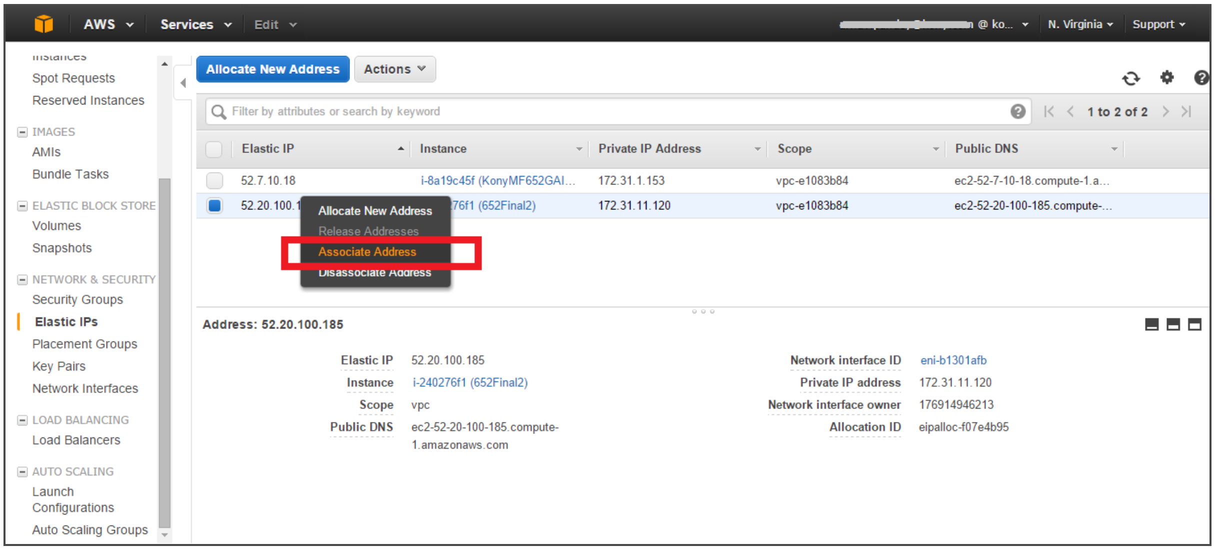
New address request succeeded
Elastic IP: 52.3.135.228. View Elastic IP

Close

Address: 52.20.100.185

Elastic IP	5	Network interface ID	eni-b1301afb
Instance	i-... (652...)	Private IP address	172.31.11.120
Scope	vpc	Network interface owner	176914946213
Public DNS	ec2-52-20-100-185.compute-1.amazonaws.com	Allocation ID	eipalloc-f07e4b95

3. Associate this IP with your instance.

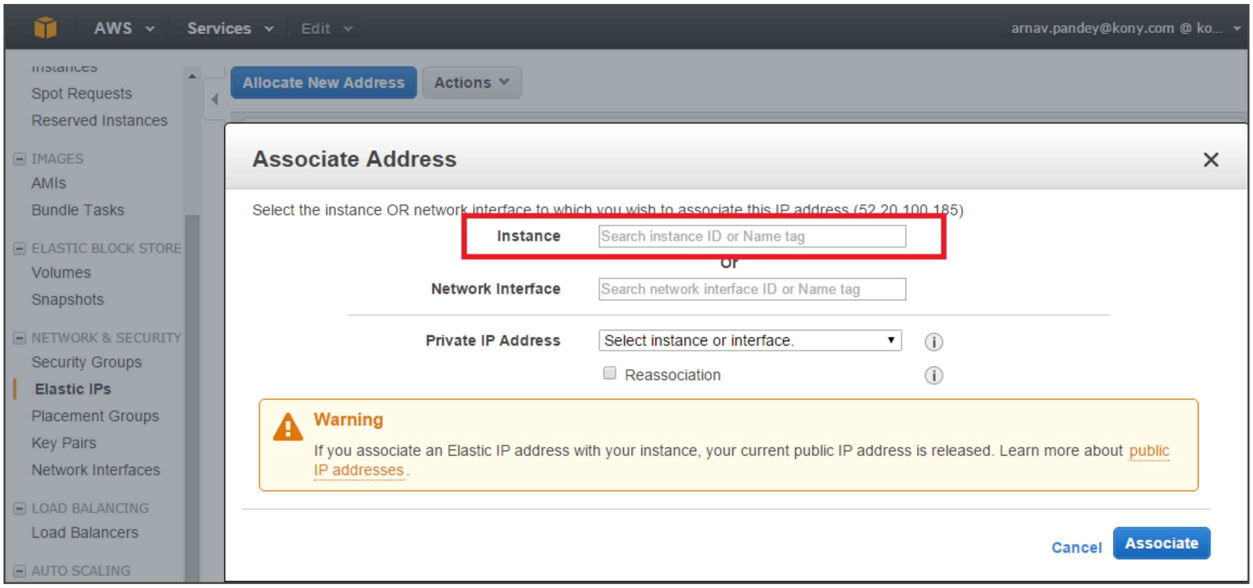


Elastic IP	Instance	Private IP Address	Scope	Public DNS
52.7.10.18	i-8a19c45f (KonyMF652GAI...	172.31.1.153	vpc-e1083b84	ec2-52-7-10-18.compute-1.a...
52.20.100.1	...76f1 (652Final2)	172.31.11.120	vpc-e1083b84	ec2-52-20-100-185.compute-...

Address: 52.20.100.185

Elastic IP	52.20.100.185	Network interface ID	eni-b1301afb
Instance	i-240276f1 (652Final2)	Private IP address	172.31.11.120
Scope	vpc	Network interface owner	176914946213
Public DNS	ec2-52-20-100-185.compute-1.amazonaws.com	Allocation ID	eipalloc-f07e4b95

4. Select the instance.



Associate Address

Select the instance OR network interface to which you wish to associate this IP address (52.20.100.185)

Instance: Search instance ID or Name tag

Or

Network Interface: Search network interface ID or Name tag

Private IP Address: Select instance or interface.

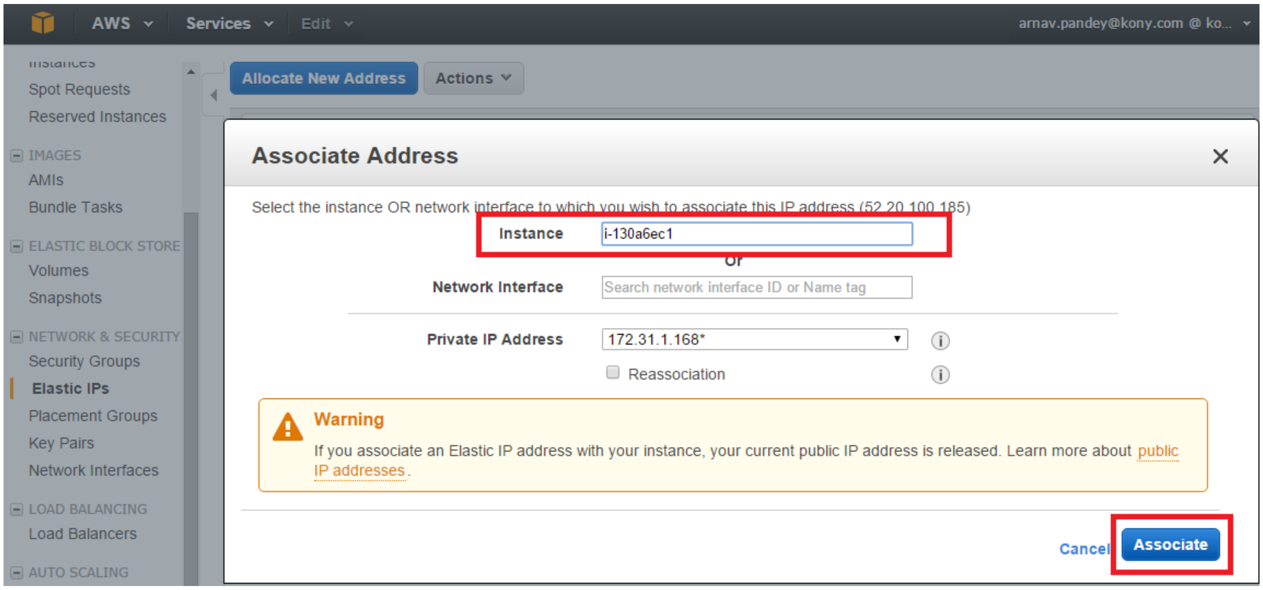
□ Reassociation

Warning

If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more about public IP addresses.

Cancel | Associate

5. Click **Associate**.



Associate Address

Select the instance OR network interface to which you wish to associate this IP address (52.20.100.185)

Instance: i-130a6ec1

or

Network Interface: Search network interface ID or Name tag

Private IP Address: 172.31.1.168*

□ Reassociation

Warning

If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more about public IP addresses.

2.7 Troubleshooting

This section lists the troubleshooting tips to resolve problems that you may encounter during installation and post installation.

2.7.1 Location for Installed MobileFabric Components

The following are the locations for installed MobileFabric components:

- **Console and Integration:**: `/home/ec2-user/KonyMobileFabric/Console`
- **Messaging**: `/home/ec2-user/KonyMobileFabric/Messaging`
- **Sync**: `/home/ec2-user/KonyMobileFabric/Sync`

2.7.2 Log Locations

The following are the log locations for installed MobileFabric components:

Console and Integration

- **Product Logs**: /home/ec2-user/KonyMobileFabric/Console/logs
- **Server Logs**: /home/ec2-user/KonyMobileFabric/Console/tomcat/logs
- **Template Logs**: /home/ec2-user/KonyMobileFabric/templatetools

Messaging

- **Product Logs**: /home/ec2-user/KonyMobileFabric/Messaging/logs
- **Server Logs**: /home/ec2-user/KonyMobileFabric/Messaging/tomcat/instance1/logs

Sync

- **Product Logs**: /home/ec2-user/KonyMobileFabric/Sync/synclogs
- **Server Logs**: /home/ec2-user/KonyMobileFabric/Sync/apache-tomcat-7.0.52/logs

2.7.3 How to Start MobileFabric Components

2.7.3.1 Starting JMS

1. To start JMS, go to /home/ec2-user/KonyMobileFabric/Console/jboss_jms/bin folder.
2. Execute the following command:

```
nohup ./startjms.sh &
```

2.7.3.2 Starting Console Manually

1. Go to `/home/ec2-user/KonyMobileFabric/Console/tomcat/bin` folder.

2. Execute the following command:

```
nohup ./startup.sh &
```

2.7.3.3 Starting Messaging Services Manually

1. Go to `/home/ec2-user/KonyMobileFabric/Messaging/tomcat/instance1/bin` folder.

2. Execute the following command:

```
nohup ./startup.sh &
```

2.7.3.4 Starting Sync Services Manually

1. Go to `/home/ec2-user/KonyMobileFabric/Sync/apache-tomcat-7.0.52/bin` folder.

2. Execute the following command:

```
nohup ./startup.sh &
```

3. How to Add Users

Identity is a service that validates the authentication of the users before accessing your application. Kony MobileFabric allows Kony User Repository for validations.

- Kony User Repository

You can setup an identity service based on the type of the users who are allowed to access your application. To allow access to your application to a larger audience, use Kony User Repository authentication.

> ***Note:*** Setting up an identity service is optional. You may choose not to implement any authentication services for your application.

3.1 Kony User Repository

With Kony User Repository, you can create log-in credentials for app users. Typically, you use the service to restrict the number of users accessing the application.

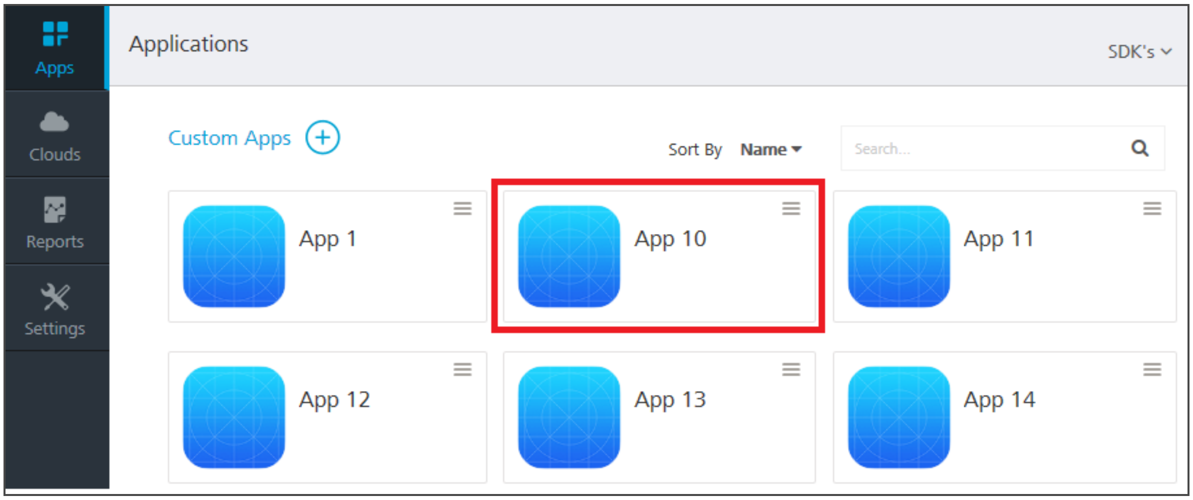
Steps to enable a user to access your application include:

- Enabling Kony User Repository Service for Your Application.
- Adding a User to the Kony User Repository

3.1.1 Enabling Kony User Repository Service for Your Application

To enable users to access your application, follow these steps:

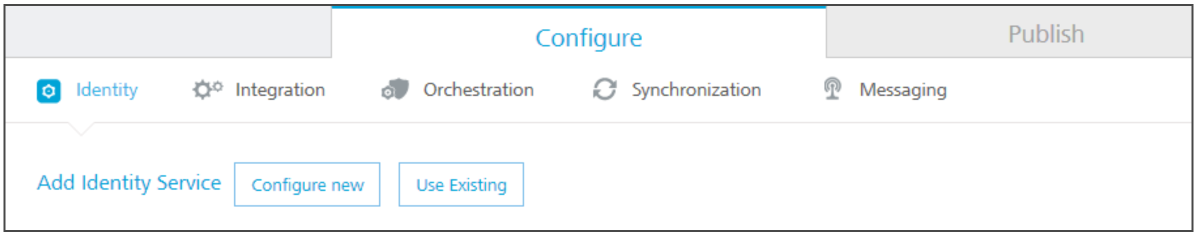
1. Create an app.
2. From the **Applications** page, click an application.



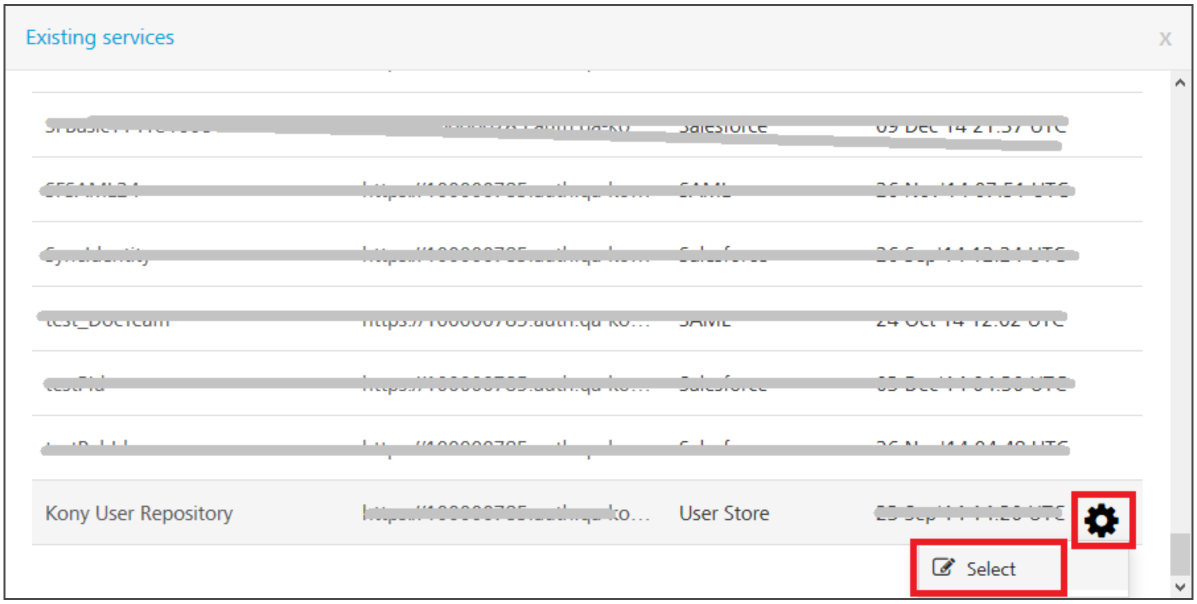
The **Identity** page appears.

> *Note:* If you want to add Kony User Repository to a new app, click **Custom Apps**.

3. Click **Add Identity Services** > **Use Existing**.



4. In the **Existing services** page, hover over the required **Kony User Repository** from the list, click the **Settings** button, and then click **Select.** The Kony User Repository is added to your app.



Existing services

x

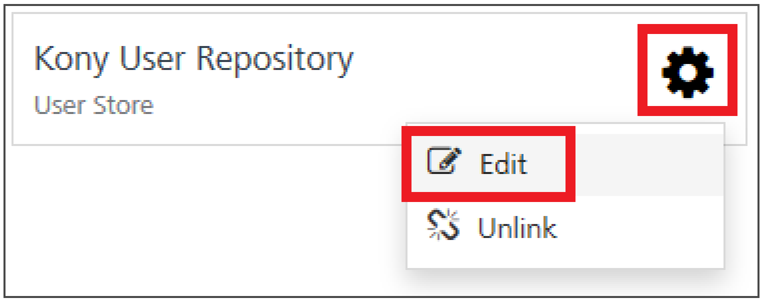
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
Kony User Repository	[redacted]ko…	User Store	[redacted]

> ***Note:*** If a **Kony User Repository** service is already added to your app, then the **Kony User Repository** service is listed in the **Existing Services** list. You can add only one Kony User Repository for an app.

3.1.2 Adding a User to Kony User Repository

To create a User Repository account for a user, follow these steps:

1. From the **Identity** tab of an app, click the **Settings** ⚙ button next to the **Kony User Repository**, and then click **Edit**.

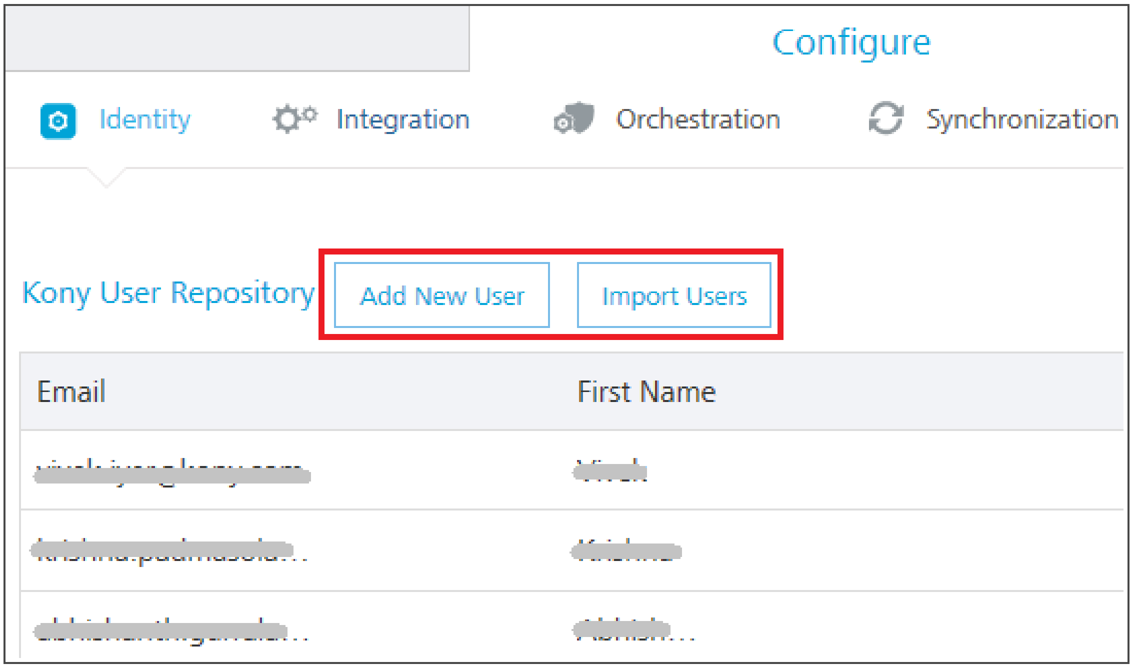


Kony User Repository

User Store

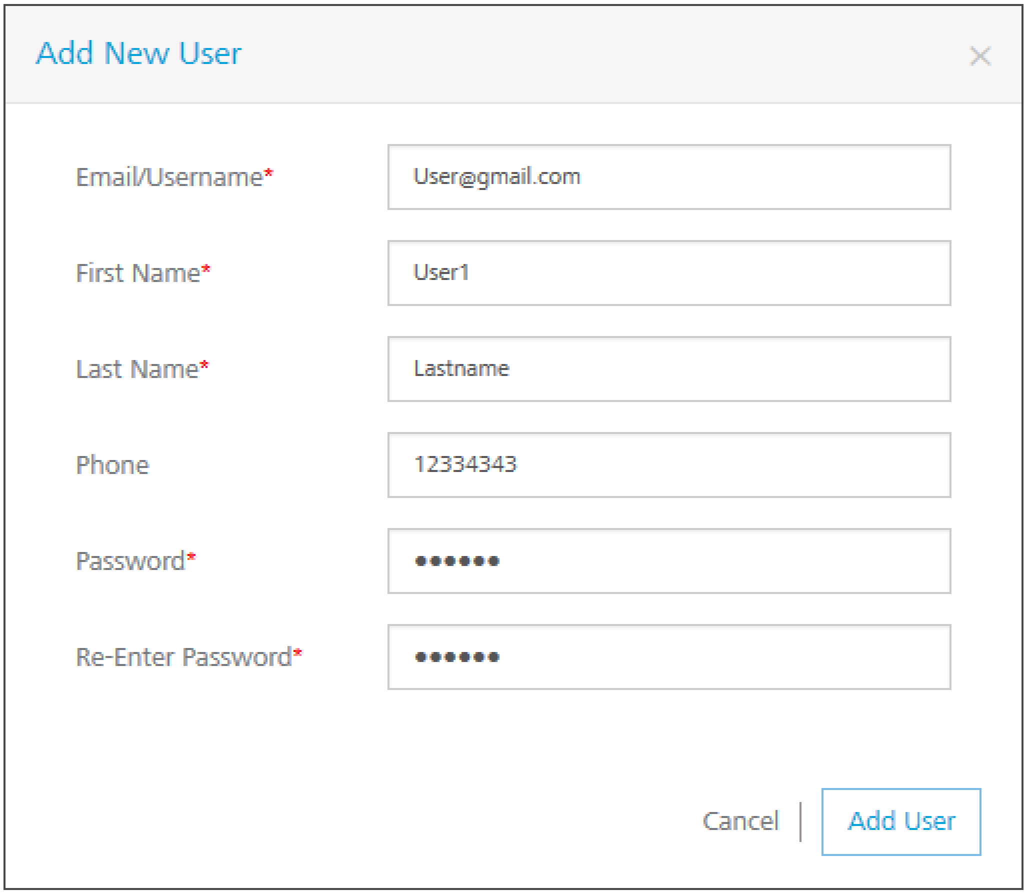
A list of existing users is displayed.

2. Click **Add New User**. The **Add New User** dialog appears.



Email	First Name
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]

3. Provide the required details.



Add New User

Email/Username*	User@gmail.com
First Name*	User1
Last Name*	Lastname
Phone	12334343
Password*	••••••
Re-Enter Password*	••••••

4. Click **Add User**.

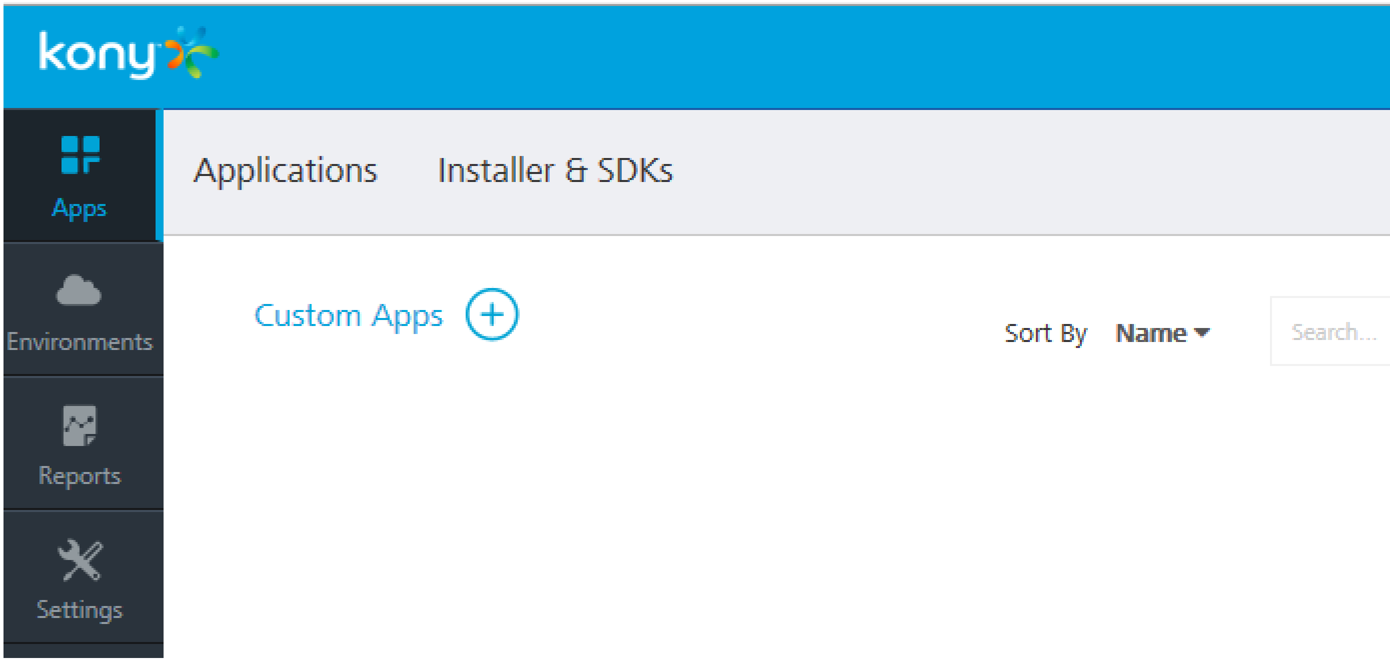
You can add another user by repeating the steps above in the procedure.

4. Creating Your MobileFabric App

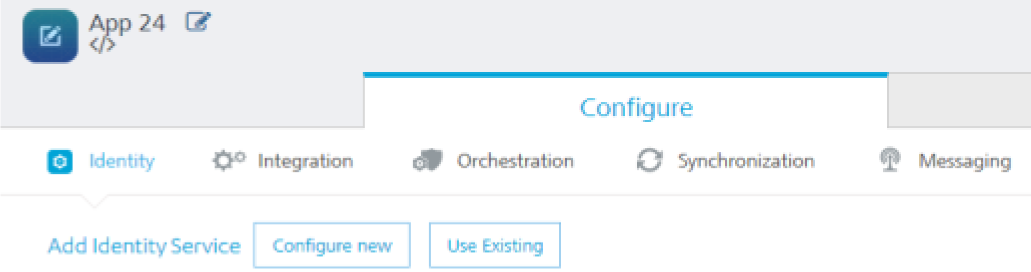
Adding a MobileFabric app to your account creates a container or a logical wrapper around all the services you want to provide for your mobile app. Once your services are published, you will receive an App Key App Secret, and Service URL, which are used within your client app development tool to securely connect to your back-end MobileFabric services. The App key, App secret, and Service URL are initialized through SDKs.

To create your MobileFabric app, follow these steps:

1. Click **Apps** from the left pane.
2. Click **Custom Apps**.



3. Rename the app and change your app icon, if required.



5. Creating an Identity Service

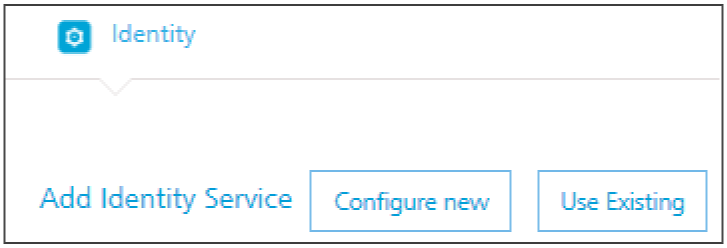
The Identity Service allows you to add a simple authentication and authorization method to your mobile app. This service can integrate with various back-end identity providers (IDP) such as Salesforce, SAP, Active Directory (direct or by Active Directory Federation Services (AD FS)) or any SAML 2.0 enabled end-point. After successfully authenticating, the back-end IDP returns a security token that is held by Kony MobileFabric. This token can be used in subsequent calls to integration or orchestration services automatically. This helps remove the burden of single sign-on token management within the client app.

For this Quick Start guide, we will use the built-in Kony User Repository provided by MobileFabric as our identity provider. For more information on integrating with other back-end IDPs, please refer to our tutorial for integrating a sample CRM app with Salesforce.

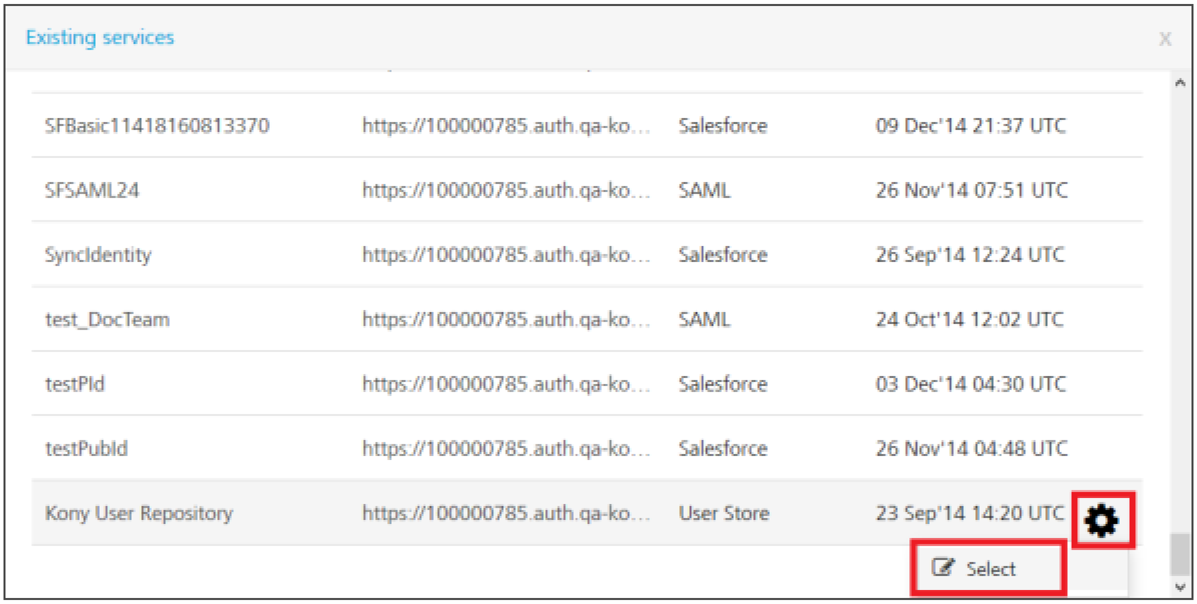
5.1 Configuring the Service

To configure an Identity service, follow these steps:

1. Click **Add Identity Service** > **Use Existing**.



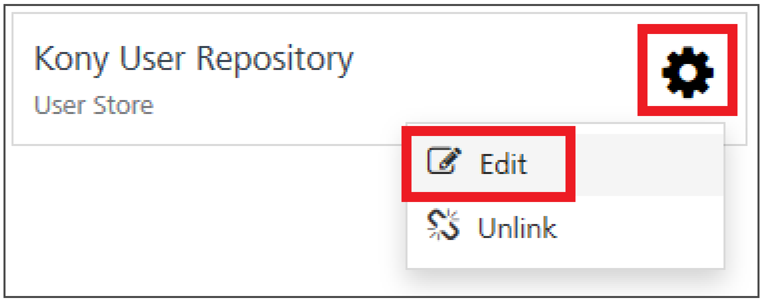
2. In the **Existing services** page, hover over the **Kony User Repository** from the list, click the **Settings** button, and then click **Select**. The user store is added to your app.



SFBasic11418160813370	https://100000785.auth.qa-ko...	Salesforce	09 Dec'14 21:37 UTC
SFSAML24	https://100000785.auth.qa-ko...	SAML	26 Nov'14 07:51 UTC
SyncIdentity	https://100000785.auth.qa-ko...	Salesforce	26 Sep'14 12:24 UTC
test_DocTeam	https://100000785.auth.qa-ko...	SAML	24 Oct'14 12:02 UTC
testPId	https://100000785.auth.qa-ko...	Salesforce	03 Dec'14 04:30 UTC
testPubId	https://100000785.auth.qa-ko...	Salesforce	26 Nov'14 04:48 UTC
Kony User Repository	https://100000785.auth.qa-ko...	User Store	23 Sep'14 14:20 UTC

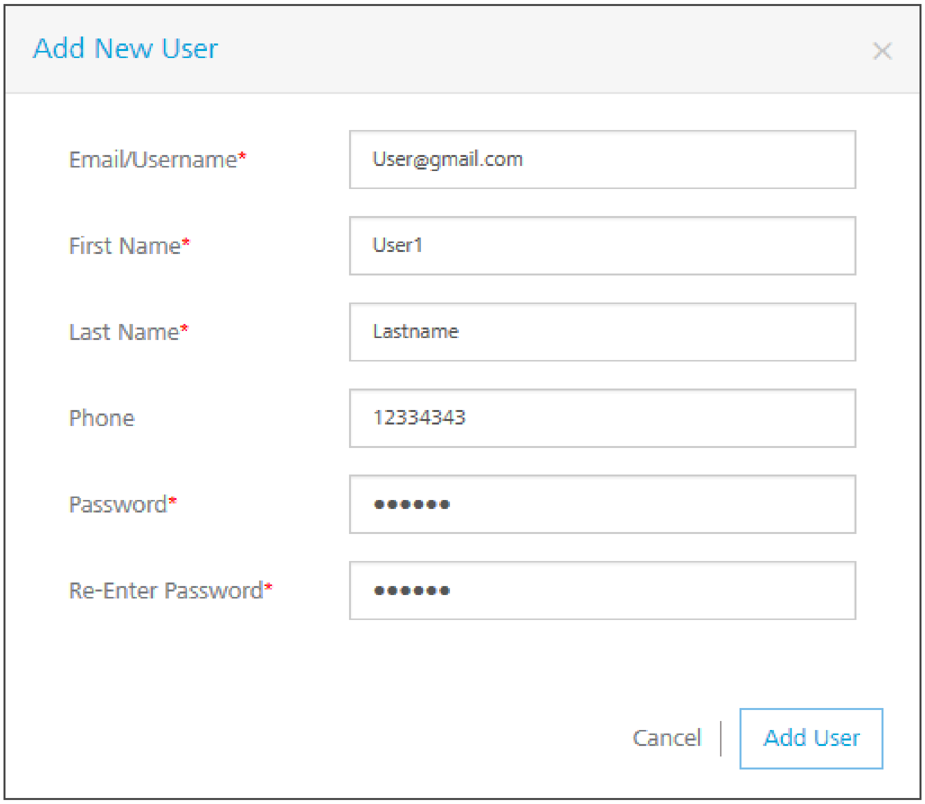
The identity service for the Kony User Repository is now available for use.

3. Click the settings icon and then click **Edit**.



4. Click **Add New User**. The **Add New User** window appears.

5. Provide the required details, and then click **Add User**.



Add New User

Email/Username*	User@gmail.com
First Name*	User1
Last Name*	Lastname
Phone	12334343
Password*	••••••
Re-Enter Password*	••••••

6. Creating an Integration Service

Now that we can authenticate our users, we need an easy way to retrieve data from an existing back-end system. In many cases, the back-end system does not return the data in the exact format we want, and/or it returns more data than our app needs.

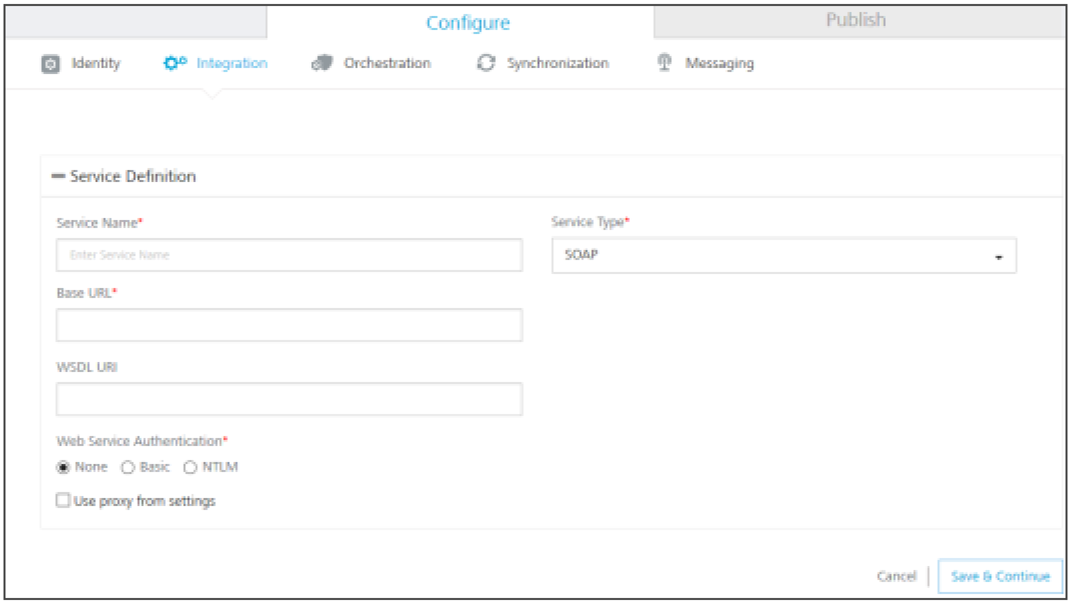
The MobileFabric Integration Services can consume data from any back- end system. You can use our standard technology connectors for REST, JSON, or SOAP web services. You can also use our enterprise business connectors that make it easy to connect to enterprise back-end systems like Salesforce or SAP, and browse for the data objects and services you want to expose to your app.

For this example, we will use a publicly available SOAP web service for getting weather information. To get the current weather and the weather forecast for a ZIP code, we will have to call two separate services.

To call the weather services, follow these steps:

1. Click the **Integration** tab.

2. Click **Add Integration** > **Configure New**.



— Service Definition

Service Name*
Enter Service Name

Service Type*
SOAP

Base URL*

WSDL URI

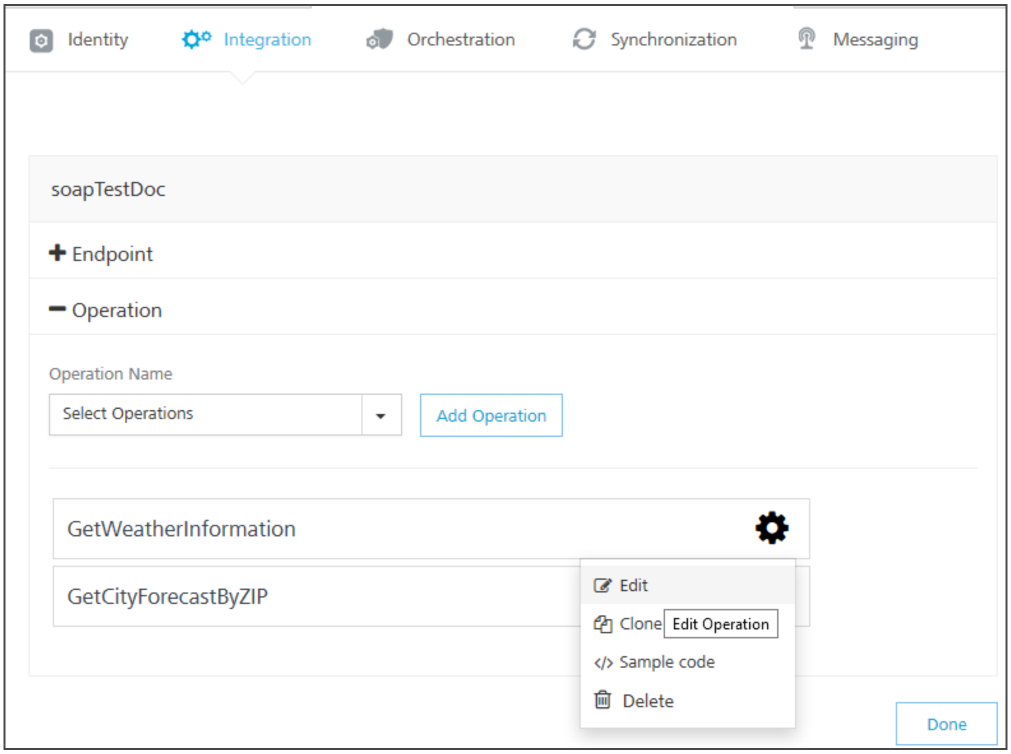
Web Service Authentication*
● None ○ Basic ○ NTLM

□ Use proxy from settings

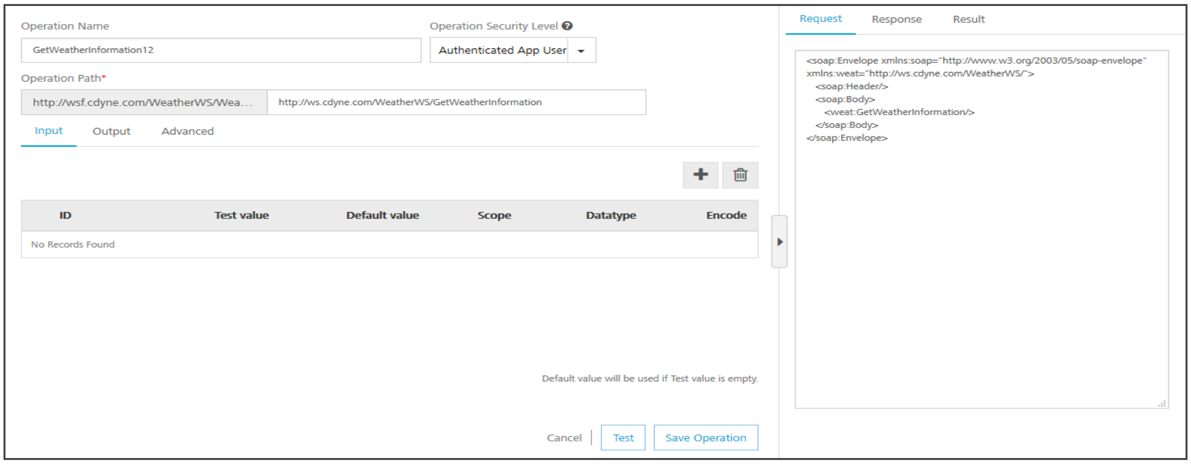
3. Name your new service **Weather** and choose *SOAP* as the **Endpoint Type**.

4. Then enter the following URLs to complete the service definition:
    - In the **Base URL** box, type: http://wsf.cdyne.com/WeatherWS/Weather.asmx
    - In the **WSDL URL** box, type: http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl
5. Under the **Web Service Authentication**, select one of the following modes:
    1. **None**: Select this option if you do not want to provide any authentication for the service.
    2. **Basic**: Provide User ID and Password if the external Web service requires form or basic authentication.
    3. **NTLM**: Your service follows the NT LAN Manager authentication process. You are required to provide the User ID, Password, NTLM Host, and NTLM Domain.
    4. To enable the proxy, select the **Use proxy from settings** check box. By default, the check box is cleared.
6. Click **Save & Continue** to retrieve the WSDL. Each of the available operations are listed in a drop-down box. Select the *GetCityForecastByZip* and *GetCityWeatherByZip*. Click **Add Operation**. This will create two operations under your Weather service that maps to the SOAP web service methods.

7. To test and edit the *GetCityWeatherByZip*, click the settings icon and choose **Edit**.



8. The operation details window opens. A sample web service request is provided showing a placeholder for any input parameters. For the GetCityWeatherByZip, the ZIP is the only input parameter displayed as `<ns1:ZIP>?XXX?</ns1:ZIP>` in the sample request. At this point, we could hard code a value, but since we want our app to provide the zip code, we need to provide an input variable name: `<ns1:ZIP>$zip</ns1:ZIP>`. We then need to define that variable under the input tab including a test value of *10036*.



Operation Name: GetWeatherInformation12

Operation Security Level: Authenticated App User

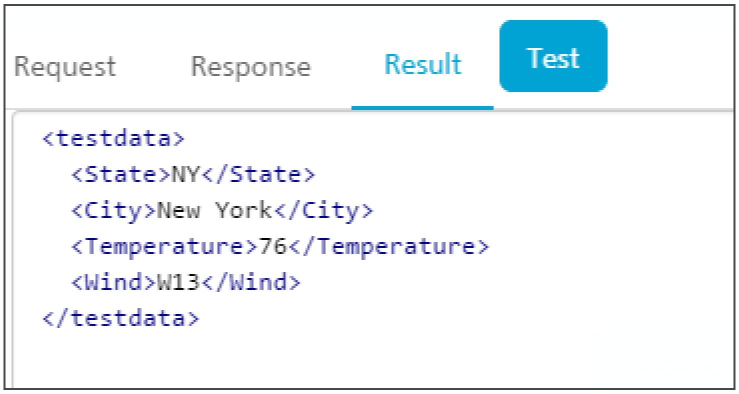
Operation Path*: http://wsf.cdyne.com/WeatherWS/Wea… http://ws.cdyne.com/WeatherWS/GetWeatherInformation

ID	Test value	Default value	Scope	Datatype	Encode
No Records Found					

Default value will be used if Test value is empty.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:weat="http://ws.cdyne.com/WeatherWS/">
  <soap:Header/>
  <soap:Body>
    <weat:GetWeatherInformation/>
  </soap:Body>
</soap:Envelope>
```

9. You can then test the service and see the SOAP web service response.



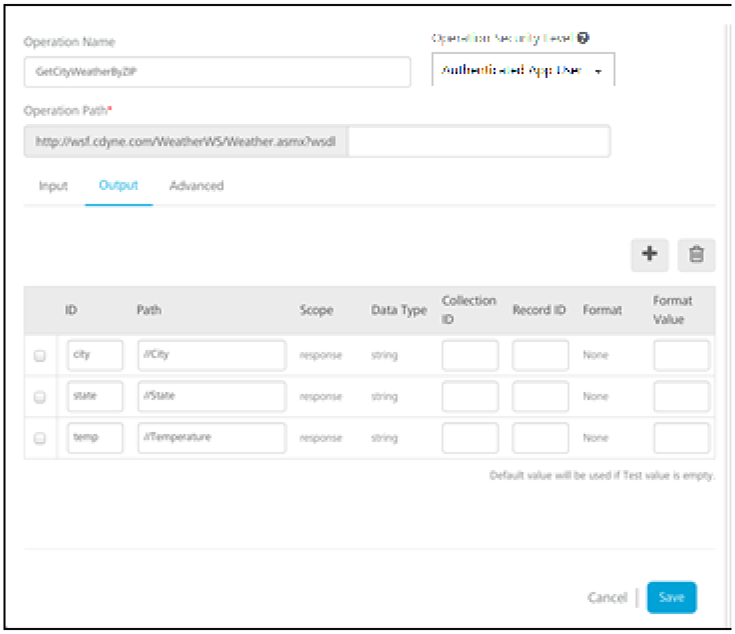
```
<testdata>
  <State>NY</State>
  <City>New York</City>
  <Temperature>76</Temperature>
  <Wind>W13</Wind>
</testdata>
```

10. On the **Output** tab, enter the following parameters and path.

city //City

state //State

temp //Temperature



Operation Name: GetCityWeatherByZIP

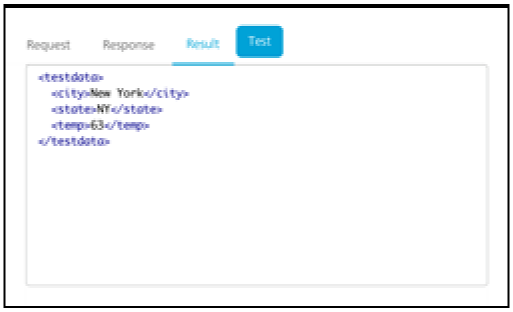
Operation Security Level: Authenticated App User

Operation Path*: http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl

	ID	Path	Scope	Data Type	Collection ID	Record ID	Format	Format Value
□	city	//City	response	string			None	
□	state	//State	response	string			None	
□	temp	//Temperature	response	string			None	

Default value will be used if Test value is empty.

11. Click the **Test** button again and the result will be displayed as XML.



```
<testdata>
  <city>New York</city>
  <state>NY</state>
  <temp>63</temp>
</testdata>
```

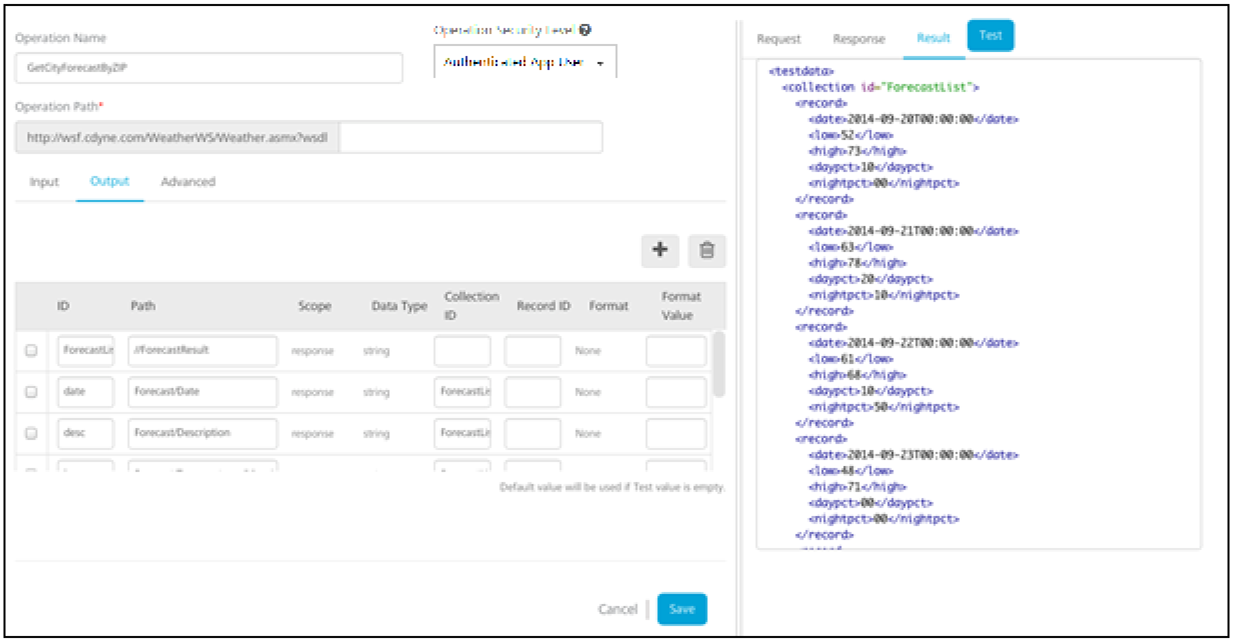
The result will be converted to a JSON before being sent to the device.

> ***Note:*** This is a simple example, but it shows the power and flexibility the MobileFabric Integration Service provides to retrieve data and process it before returning an optimized JSON string to the device. You can also configure additional processing under the advanced tab including deploying custom code that executes before and after the service invocation.

12. Edit the *GetCityForecastByZip* in the same way. This service returns a repeating data structure for each day providing that day's weather forecast. This requires the use of the collection ID under the output tab to create a repeating set of JSON objects. After creating the **ZIP** input parameter the same way as the previous service, enter the following output parameters:

ID	xPath	CollectionID
ForecastList	//ForecastResult/Forecast	
date	Forecast/Date	ForecastList
desc	Forecast/Description	ForecastList
low	Forecast/Temperatures/MorningLow	ForecastList
high	Forecast/Temperatures/DaytimeHigh	ForecastList
daypct	Forecast/ProbabilityOfPrecipiation/Daytime	ForecastList
nightpct	Forecast/ProbabilityOfPrecipiation/Nighttime	ForecastList

Test your service and you will see the resulting XML showing the repeating collections of forecasts.



Operation Name: GetCityForecastByZIP

Operation Security Level: Authenticated App User

Operation Path*: http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl

	ID	Path	Scope	Data Type	Collection ID	Record ID	Format	Format Value
□	ForecastLi	//ForecastResult	response	string			None	
□	date	Forecast/Date	response	string	ForecastLi		None	
□	desc	Forecast/Description	response	string	ForecastLi		None	

Default value will be used if Test value is empty.

```
<testdata>
  <collection id="ForecastList">
    <record>
      <date>2014-09-20T00:00:00</date>
      <low>52</low>
      <high>73</high>
      <daypct>10</daypct>
      <nightpct>00</nightpct>
    </record>
    <record>
      <date>2014-09-21T00:00:00</date>
      <low>63</low>
      <high>78</high>
      <daypct>20</daypct>
      <nightpct>10</nightpct>
    </record>
    <record>
      <date>2014-09-22T00:00:00</date>
      <low>61</low>
      <high>68</high>
      <daypct>10</daypct>
      <nightpct>50</nightpct>
    </record>
    <record>
      <date>2014-09-23T00:00:00</date>
      <low>48</low>
      <high>71</high>
      <daypct>00</daypct>
      <nightpct>00</nightpct>
    </record>
```

7. Creating an Orchestration Service

The following types of Orchestration Services are supported by MobileFabric:

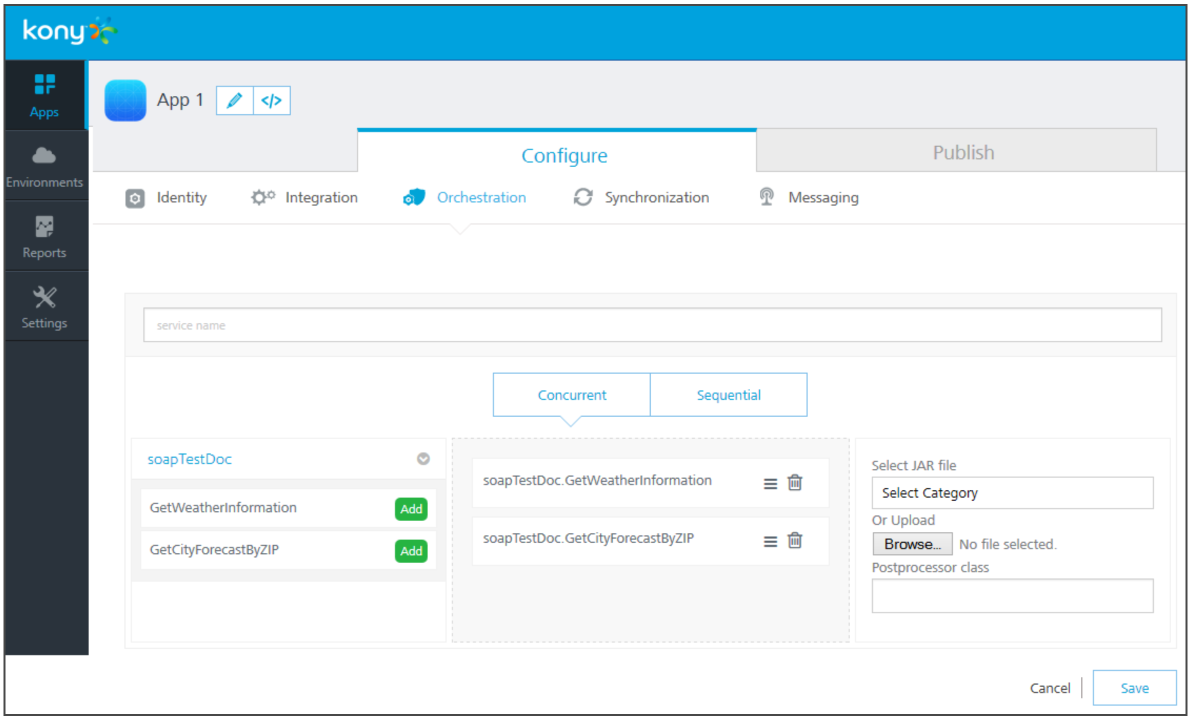
- **Composite Services** include:
    - **Concurrent Service**: All specified integration services are called in parallel.
    - **Sequential Service**: The output of one service can be used as the input of a subsequent service. Each integration service in the chain will be executed sequentially.
- **Looping Service**: Allows you to call the same service in a loop using the same input values until you reach a break condition or you can send in a delimited set of input values and the service will loop through the inputs until it reaches the end.

For this example, we want to return the current weather and the forecast in one service call. Therefore, we want to create a concurrent composite Orchestration Service using our *GetCityWeatherByZip* and our *GetCityForecastByZip*. This allows our app to call one Orchestration Service using the ZIP Code and getting back all the data we need.

To execute an Orchestration Service, follow these steps:

1. On the **Orchestration** tab, click **Add Orchestration** > **Create Composite**.
2. Enter a name for the service as **GetCityWeatherAndForecastByZip** and choose **Concurrent** as the type of Composite service.

3. Add the *GetCityWeatherByZip* and the *GetCityForecastByZip* operations.



4. Save the service.

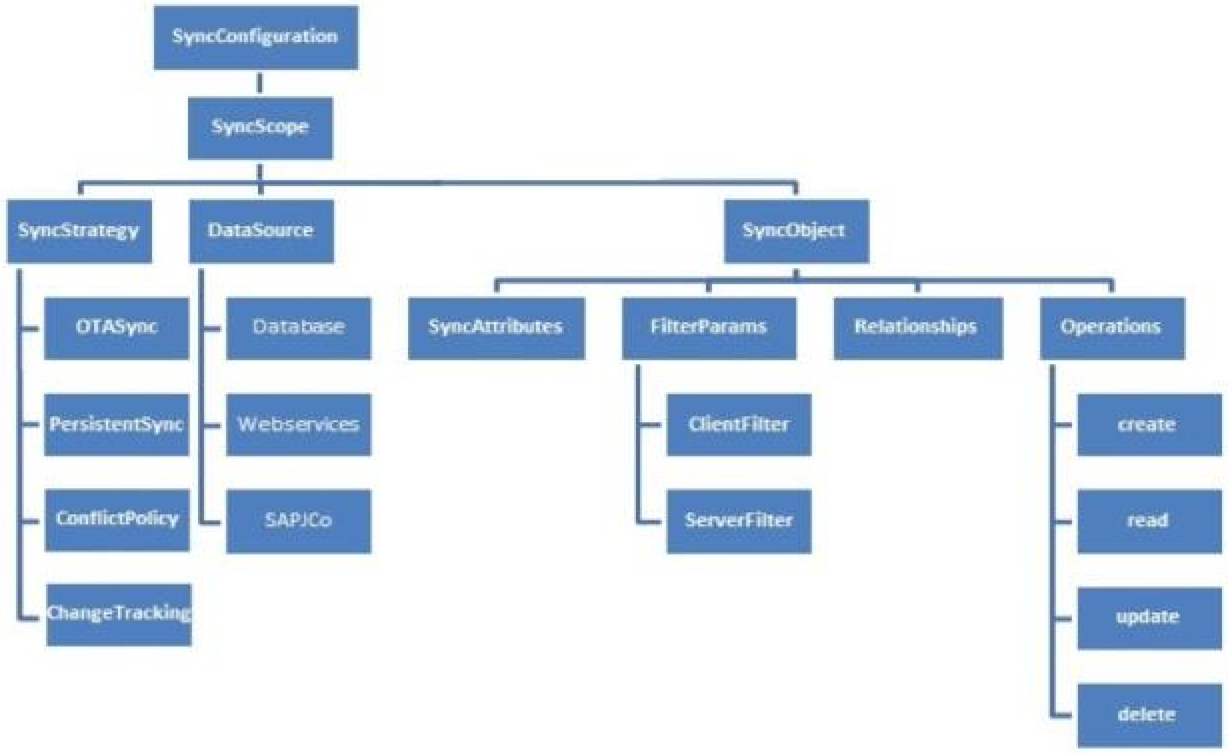
8. Synchronization

Kony Synchronization is a comprehensive data synchronization platform that enables developers to add synchronization capabilities to mobile applications. Fundamental to Sync Framework is the ability to support offline and collaboration of data between devices and the backend systems.

To enable synchronization capability for an app, you need to define a Sync Configuration file.

8.1 Sync Configuration file

A Sync Configuration captures details of the data synchronization characteristics of an application. These details are captured in a file typically referred to SyncConfig.xml (the name really does not matter) adhering to the SyncConfig.xsd schema. A SyncConfig.xml represents the below structure.



The two most important elements of this schema are:

- Sync Scope
- Sync Object

8.1.1 Sync Scope

A Sync Scope groups together the Sync Objects that share common synchronization characteristics like Sync Strategy, Datasource and so on.

A Sync Configuration can have multiple Sync Scopes. It is not possible to define relationships between Sync Objects belonging to different Sync Scopes.

8.1.2 Sync Object

Conceptually, you can consider a Sync Object as a business object that has some public attributes and some methods. The public attributes correspond to the fields visible to client devices, and they are used for synchronization. The methods correspond to the CRUD operations that map to the backend services exposed for the object. The parameter values methods /operations based on both public attribute values.

A Sync Object is meta-data:

- Defining the business object model of an application.
- Defining the way data is exchanged between mobile devices and backend.

A Sync Object is data:

Sync Object data is a business object instance exchanged between client and server.

8.2 Adding a New Synchronization Scope

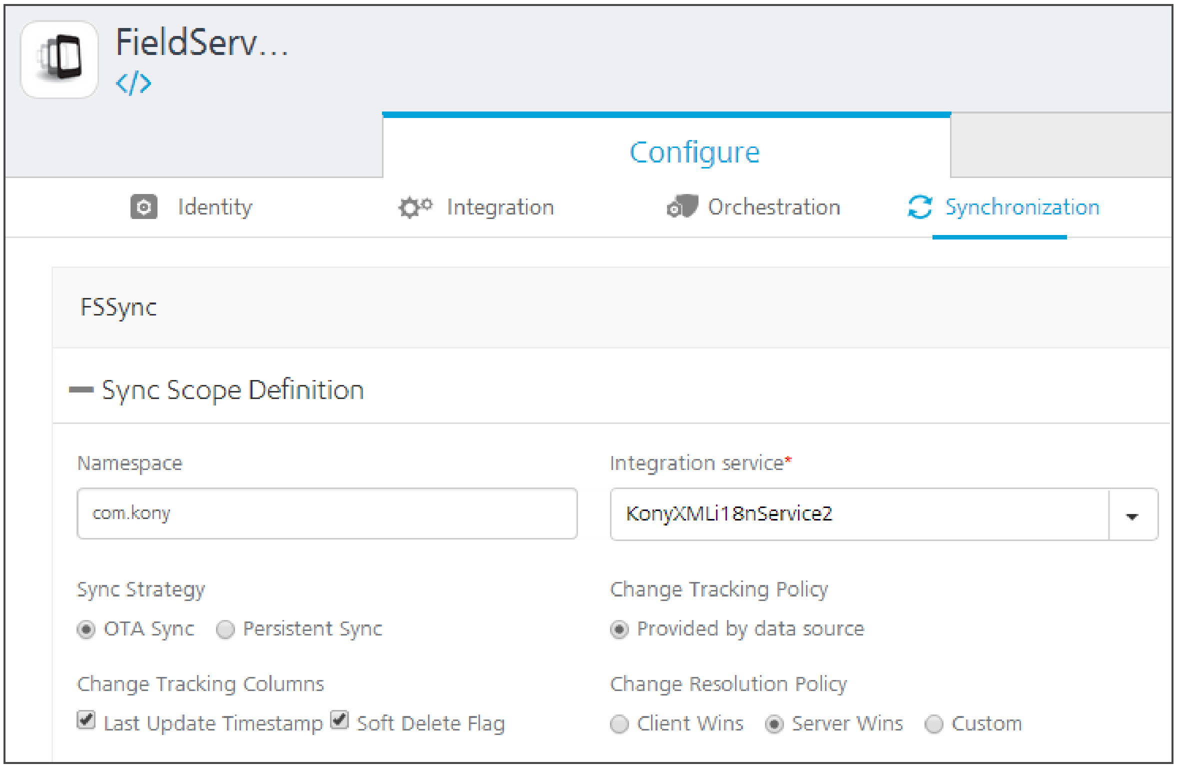
> ***Note:*** The following section explains setting up a Sync scope for Salesforce account.

To add a new Synchronization scope, follow these steps:

1. From the **Synchronization** page, click **Add Synchronization Scope** > **Configure New**.

2. Provide a name for the new Sync scope. (For example, *FSSync*)

3. Under **Sync Scope Definition**, provide the following details:



FSSync

— Sync Scope Definition

Namespace: com.kony

Integration service*: KonyXMLi18nService2

Sync Strategy: ● OTA Sync ○ Persistent Sync

Change Tracking Policy: ● Provided by data source

Change Tracking Columns: ☑ Last Update Timestamp ☑ Soft Delete Flag

Change Resolution Policy: ○ Client Wins ● Server Wins ○ Custom

a. Specify a **Namespace** for the Sync scope. The Namespace should follow a prescribed format such as *com.kony*.

b. Select the required service from the **Integration Service** list.

c. Select a **Sync Strategy**. The available options are **OTA Sync** and **Persistent Sync**.

> ***Note:*** To understand which strategy to use for your sync scope, refer section Appendix - Sync Strategy.

d. Select a Change Tracking Policy (CTP) if you want to track the changes happening in the server database. Select Provided by data source, if you have a provision to track changes in the data source. For database this would be like a timestamp column which updates for any changes made to the row. Set CTP as Kony Sync Server, if you want SyncServer to track the changes. This option will be available only if you had selected Persistent Sync as Sync Strategy.

e. In the **Change Tracking Columns**:

   i. Select the **Last Updated Timestamp** check box when you have column that represents the latest edited values.

   ii. Select the **Soft Delete Flag** check box when the database has the column that represents soft deletes. Soft delete field in a record represents that a particular record is deleted by changing the status to deleted. This record will exist in the database. Thus by selecting this field Kony Sync server does not sync records whose status is set as deleted.

f. In case of conflicts between the data at the client and server end, specify any of the following under **Change Resolution Policy**:

   - Client Wins: The changes on the client side take precedent over the changes on the server side.
   - Server Wins: The changes on the server side take precedent over the changes on the client side.
   - Custom: Enables you to upload an Interceptor class which comprises the logic or policy for conflict resolution.

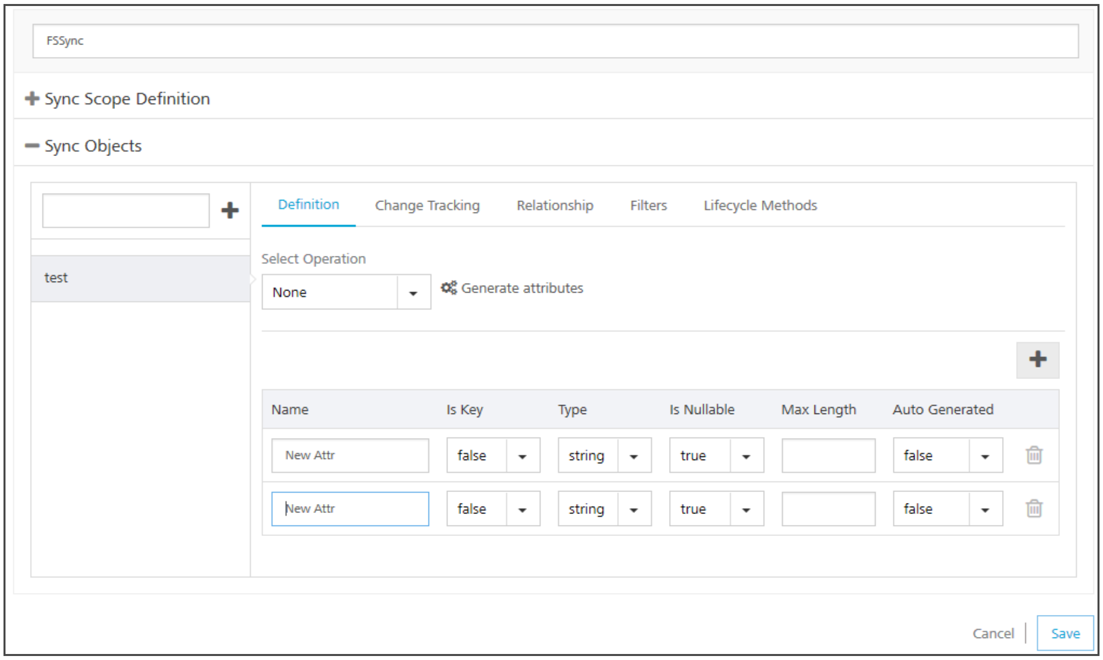
4. Expand **Sync Objects**.

5. Under **Sync Objects**, provide the following details:

   a. On the left pane, provide a name for your Sync object, and then click the **Plus** button.

   b. On the **Definition** tab of the new Sync object, select an operation from the **Select Operation** list, and click **Generate attributes**.

> ***Note:*** The list of operations available for a new Sync object depends on the Integration Service selected in the Sync Scope.



FSSync

test

Select Operation: None

Name	Is Key	Type	Is Nullable	Max Length	Auto Generated
New Attr	false	string	true		false
New Attr	false	string	true		false

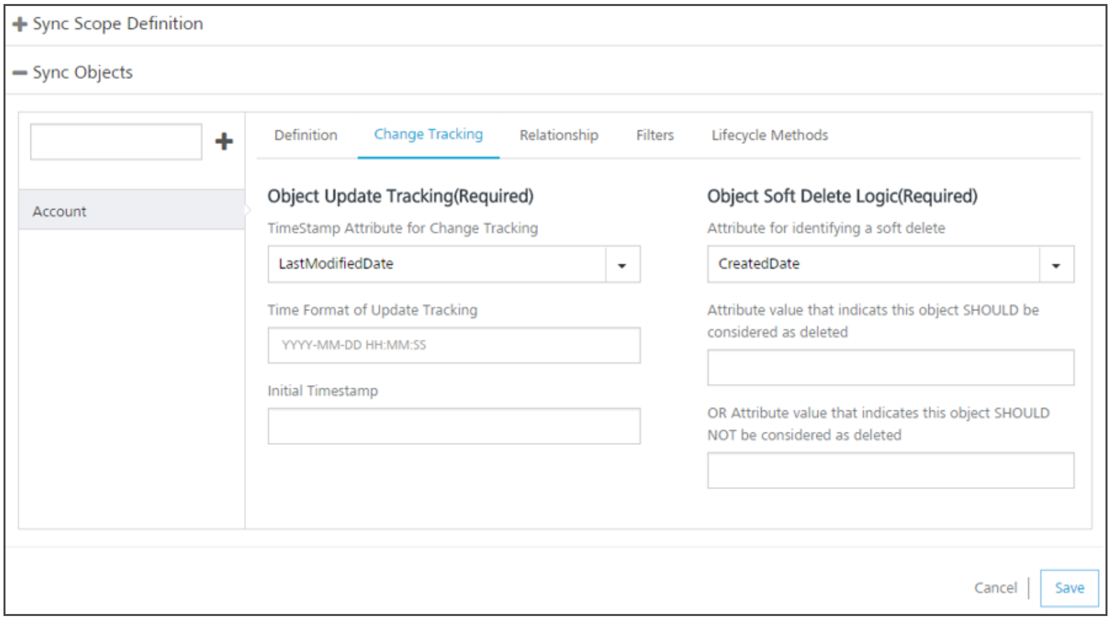
6. On the **Change Tracking** tab, do the following:

   a. From the **TimeStamp Attribute for Change Tracking** list, select an attribute that denotes a particular record is modified.

   b. From the **Attribute for Identifying a soft deleted** list, select an attribute that denotes a soft delete.

> ***Note:*** You need to select **TimeStamp Attribute for Change Tracking**, only if you have selected **Last Update Timestamp** check box under the **Change Tracking Columns** respectively.

> ***Note:*** You need to select **Attribute for Identifying a soft delete** only if you have selected **Soft Delete Flag** check box under the **Change Tracking Columns**.



Object Update Tracking(Required)

TimeStamp Attribute for Change Tracking: LastModifiedDate

Time Format of Update Tracking: YYYY-MM-DD HH:MM:SS

Initial Timestamp:

Object Soft Delete Logic(Required)

Attribute for identifying a soft delete: CreatedDate

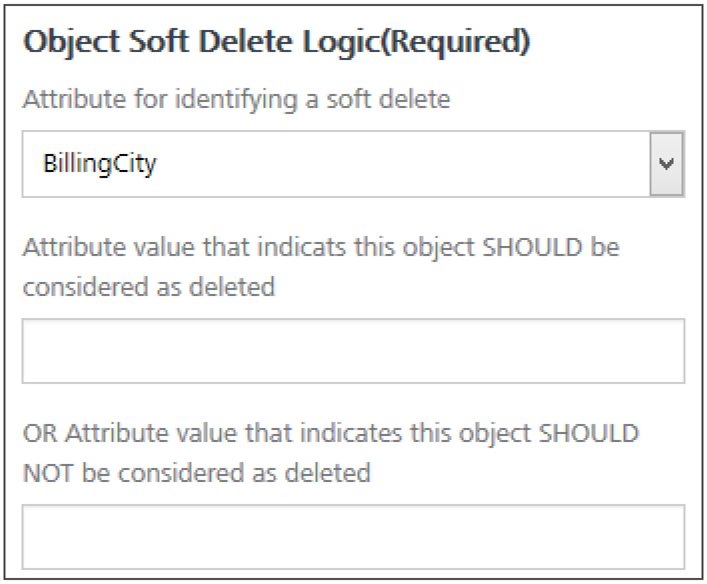
Attribute value that indicats this object SHOULD be considered as deleted:

OR Attribute value that indicates this object SHOULD NOT be considered as deleted:

For non Boolean attributes, enter additional values that will be considered for soft deleting. For example, from the list if you select **BillingCity**, the system displays the following fields.

- **Attribute value that indicates this object SHOULD be considered as deleted**: if this value matches with the main attribute, the system deletes this attribute.
- **OR Attribute value that indicates this object SHOULD NOT be considered as deleted**: if this value matches with the main attribute, the system does not delete

this attribute.



Object Soft Delete Logic(Required)

Attribute for identifying a soft delete

BillingCity

Attribute value that indicats this object SHOULD be considered as deleted

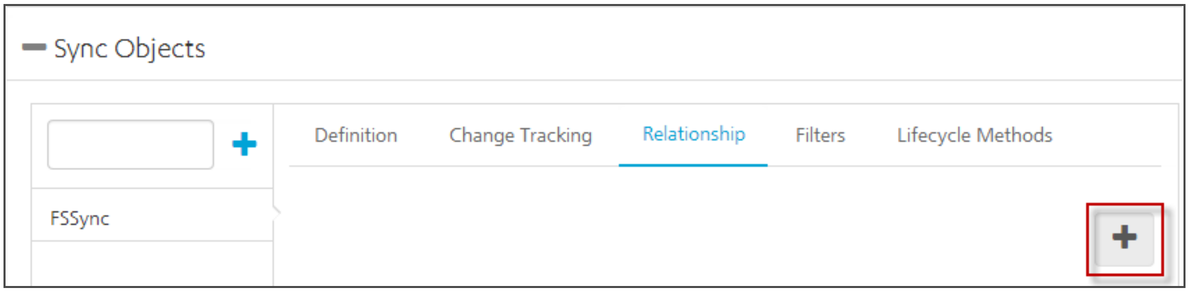
OR Attribute value that indicates this object SHOULD NOT be considered as deleted

c. Change **Time Format of Update Tracking**, if required. By default, Salesforce time format is *YYYY-MM-DD HH:MM:SS*.

d. In **Initial Timestamp** box, enter the date from which the records are to fetched.

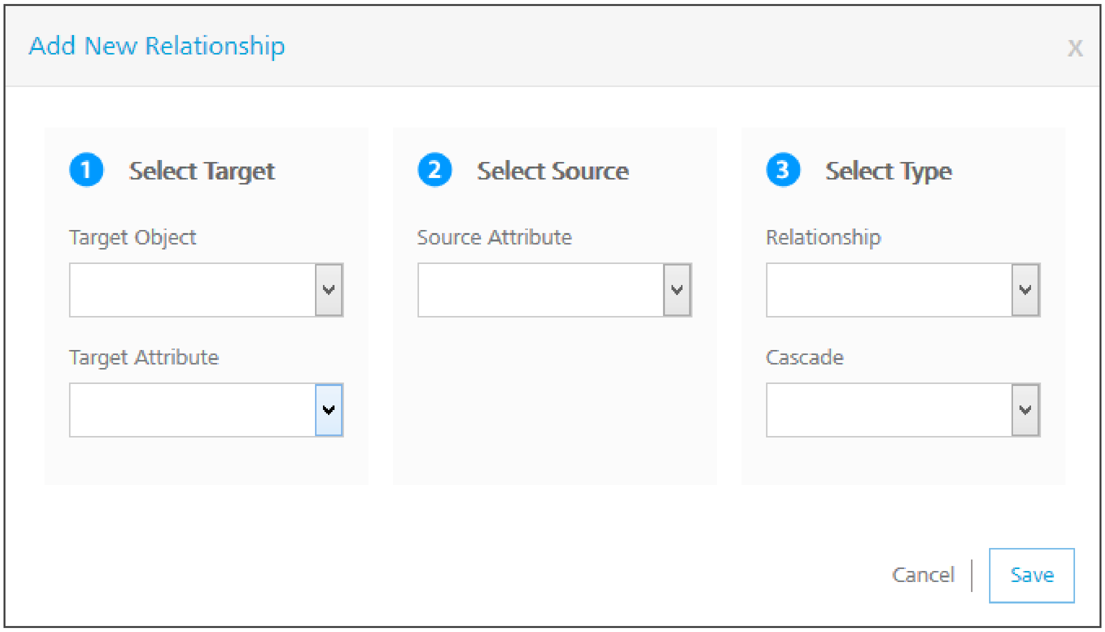
e. Click **Save**.

7. On the **Relationship** tab



Click the **Plus** button to open **Add New Relationship** dialog.

a. Provide the following details:



i. Select the required object from the **Target Object** list.

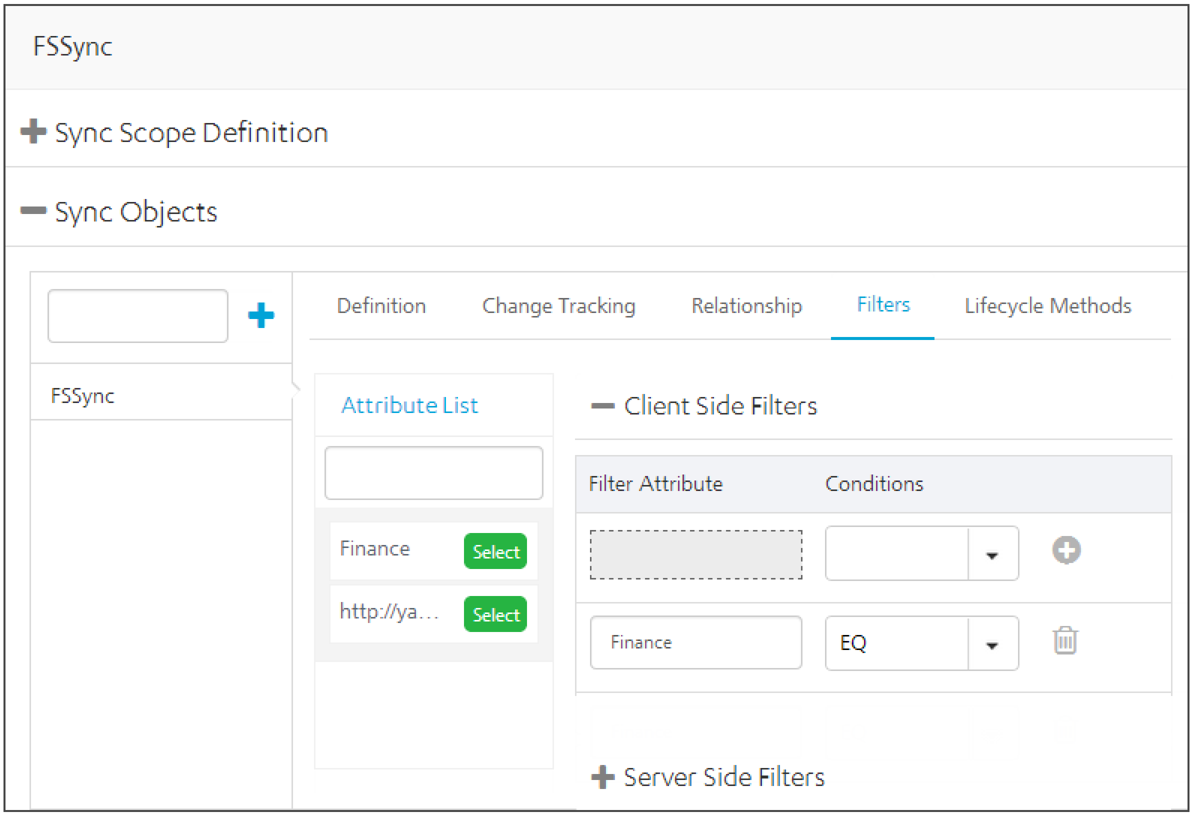
ii. Select the required attribute from the **Target Attribute** list.

iii. Select the required attribute from the **Source Attribute** list.

iv. Select the type of relation between Source attribute and target attribute from the **Relationship** list.

v. Select *True* from the **Cascade** list if you want to delete a record in the parent table and its child tables.

8. On the **Filters** tab, provide the following details:



a. In the **Client Side Filters**, from the **Attribute List**, select an attribute.

b. For the selected attribute, provide a condition.

c. To save the current filter and add another filter, click the **Plus** button.

d. In the **Server Side Filters**, from the **Attribute List**, select an attribute.

e. For the selected attribute, provide a condition.

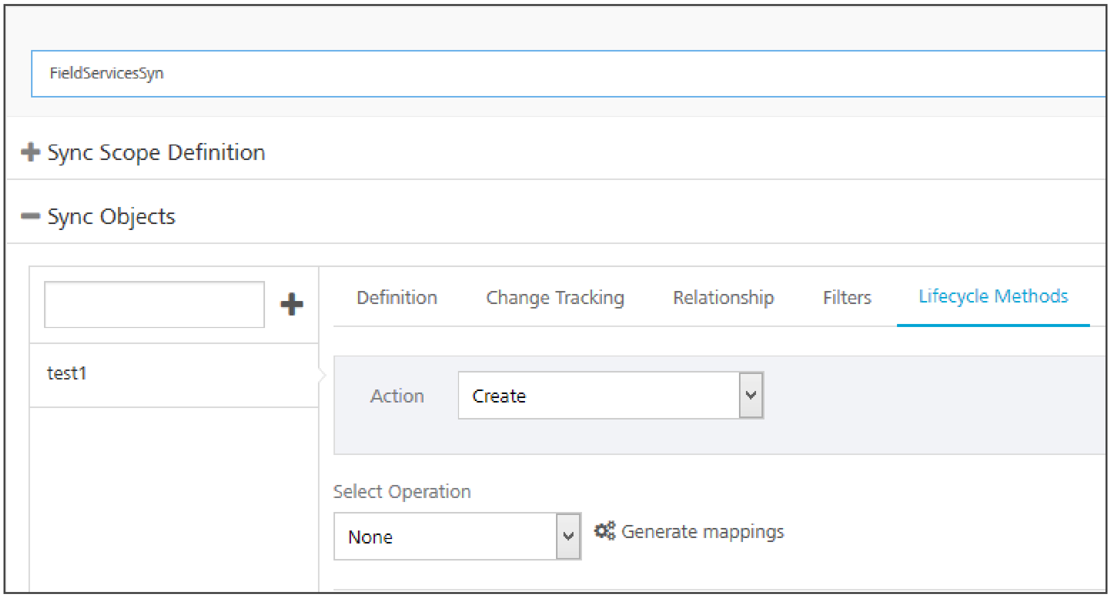
f. To save the current filter and add another filter, click the **Plus** button.

9. On the **Lifecycle Methods** tab, provide the following details:

a. From the **Action** list, select an action.

b. From the **Select Operation** list, select an operation.

c. Click **Generate Mappings**.



FieldServicesSyn

test1

Action: Create

Select Operation: None

> ***Note:*** Input mapping is generated only for *Create*, *Update* and *Delete* operations.

> ***Note:*** Output mapping is generated for all the operations: *Create*, *Update*, *Delete*, *get*, *getUpdated*, *getDeleted* and *getBatch*.

> ***Note:*** Header Mapping needs to be added manually.

d. Add Input parameters from the **Input Mapping** by clicking the **Plus** button. Provide the following details:

   a. From the **Source Type** list, select the type of the source.

   b. From the **Source Value** list, select a value.

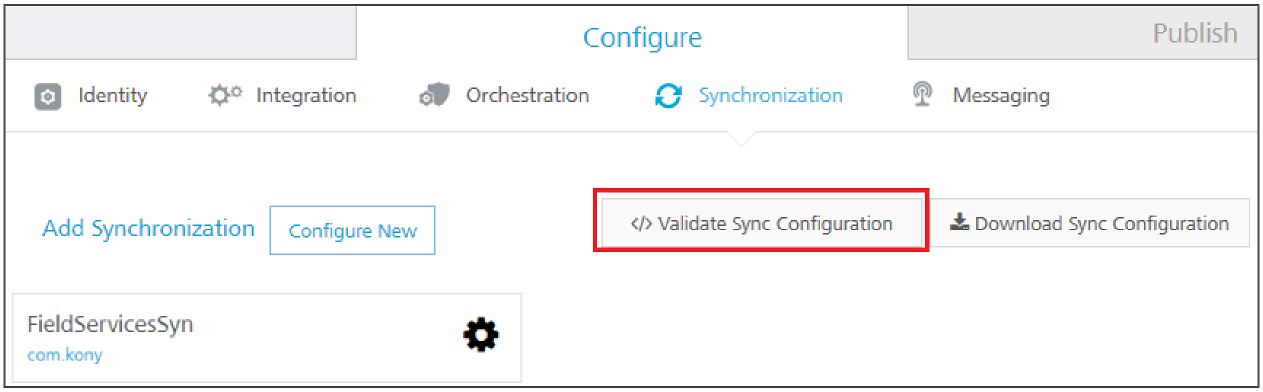
   c. From the **Service Input Param** list, select an input parameter.

   d. Click **Save**.

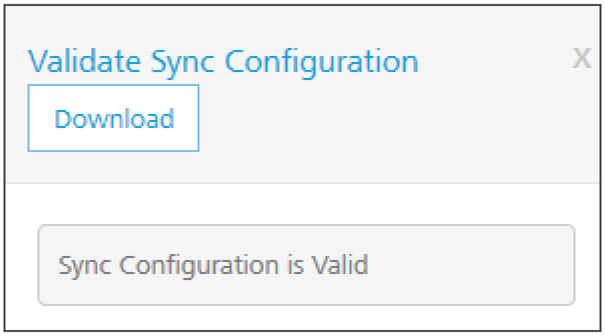
8.3 Validate Sync Configuration

Kony MobileFabric allows you to validate the Sync configuration before you can utilize the scope in your application.

To validate your Sync configuration, on the **Synchronization** page, click **Validate Sync Configuration**.



You receive the following message if your scope is valid:

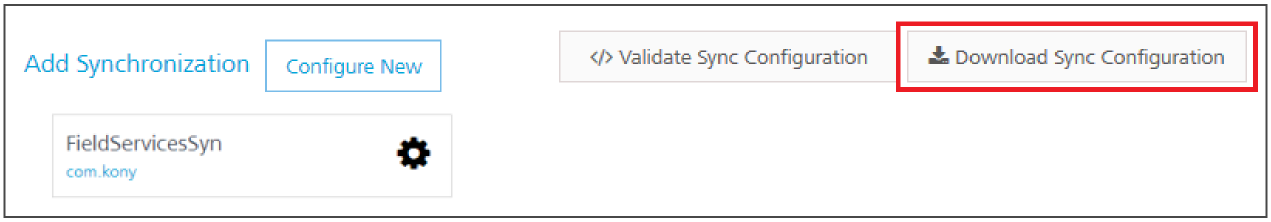


Sync Configuration is Valid

To download the file, click **Download**. This file is useful when the Sync Scope is invalid, and you wanted to know the details of the errors encountered while validating the Sync Scope.

8.4 Download the Sync Configuration

Click **Download Sync Configuration** to download the Sync configuration file *Syncconfig.xml* file on your computer.

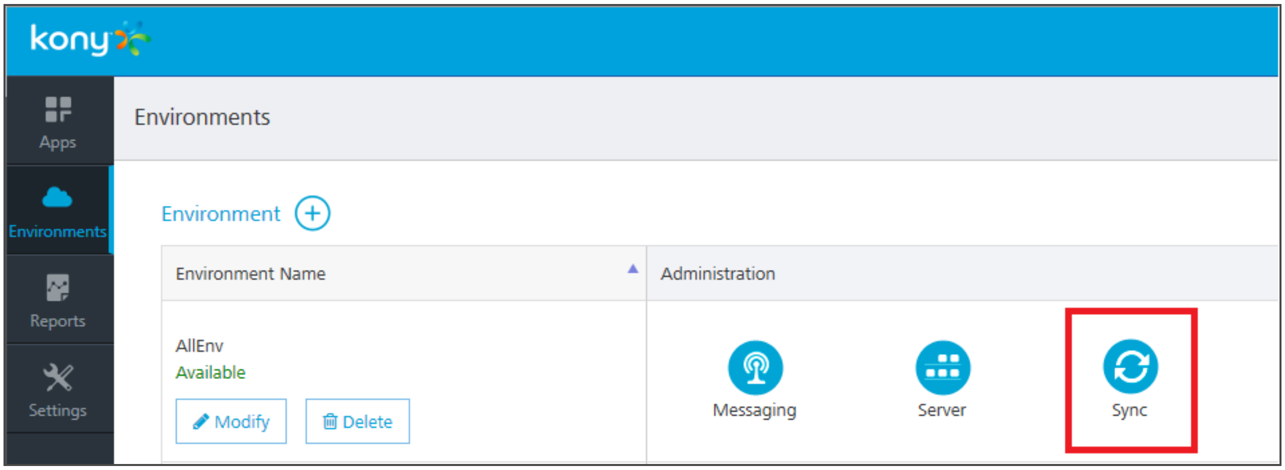


8.5 Sync Console

> ***Note:*** The details of your sync scope will be available in Sync Services after you *publish* the app.

Kony Sync Management Console provides a single point of control for monitoring and configuring the Kony sync console creation process.

To view your Sync Console, click **Sync Services** from your cloud account.



For more details on Sync Console, refer to the following document: http://docs.kony.com/konylibrary/default.htm#../Subsystems/Kony_Sync_Cloud_and_On-Premises_Console_User_Guide/Content/Homepage.htm

9. Messaging

Messaging service allows you to upload push certificates for iOS, Android, BlackBerry, and Windows 8 RT platforms.

For sending messages, follow these steps:

1. Add Push Certificates
2. Access Messaging Console
3. Send a Push Message

9.1 Add Push Certificates

MobileFabric messaging supports the following platforms:

1. iOS
2. Android
3. BlackBerry
4. Windows 8 RT & Pro

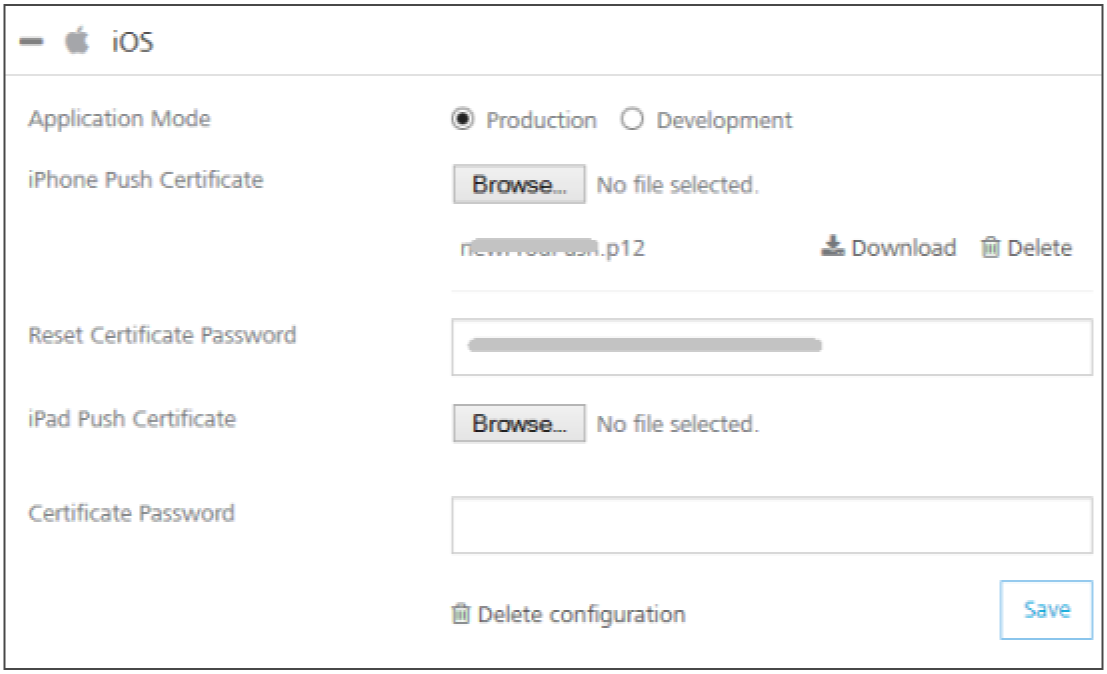
This section details the process for adding push certificates to your application.

9.1.1 iOS

> ***Note:*** Refer to the following section for creating a push certificate: http://docs.kony.com/konylibrary/default.htm#../Subsystems/KMS_Console_User_Guide/Content/Apps/Adding_Platform.htm

To add iOS Push Certificates for your app, follow these steps:

1. Expand **iOS**. A list of configurable items appear.



iOS

Application Mode ● Production ○ Development

iPhone Push Certificate Browse... No file selected.

[redacted].p12 Download Delete

Reset Certificate Password [redacted]

iPad Push Certificate Browse... No file selected.

Certificate Password

Delete configuration Save

2. **Application Mode**: An appropriate application mode.
   - Production mode: When selected, production certificates and associated password details are entered while sending push notifications. Push notifications are delivered in real-time.
   - Development mode: When selected, you can still send push message notifications, but delivery of push notifications are not real-time.
3. **iPhone Push Certificate**: From here, you can upload, download, or delete a certificate.
   - Click **Browse** to upload an iPhone certificate.
   - Click **Download** to download an iPhone certificate.
   - Click **Delete** to delete an iPhone certificate.
4. **Certificate Password**: Enter the password for iPhone, and then click Save to complete the configuration process.

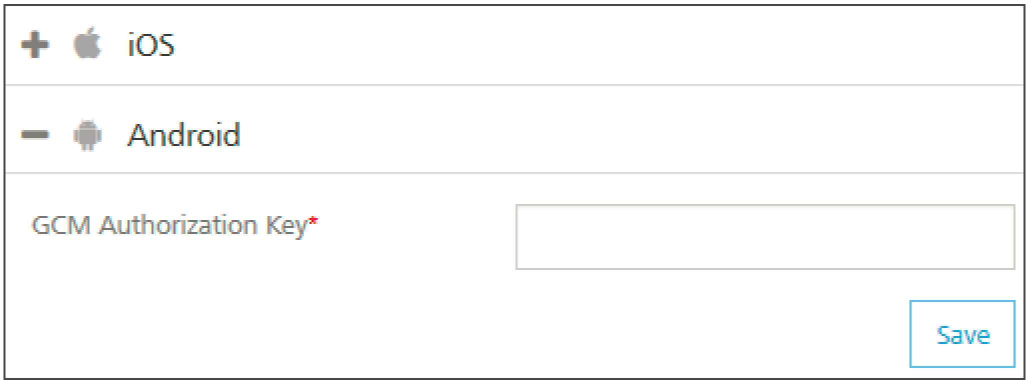
5. **iPad Push Certificate**: From here, you can upload, download, or delete a certificate.
    - Click **Browse** to upload an iPhone certificate.
    - Click **Download** to download an iPhone certificate.
    - Click **Delete** to delete an iPhone certificate.
6. Click **Save** to complete the configuration process for iOS platform.

9.1.2 Android

> ***Note:*** Refer to the following section for creating a push certificate: http://docs.kony.com/konylibrary/default.htm#../Subsystems/KMS_Console_User_Guide/Content/Apps/Adding_Platform.htm

To add Android Push Certificates for your app, follow these steps:

1. Expand **Android**. A list of configurable items appear.



2. Enter the GCM authorization key, and then click **Save** to complete the configuration process.

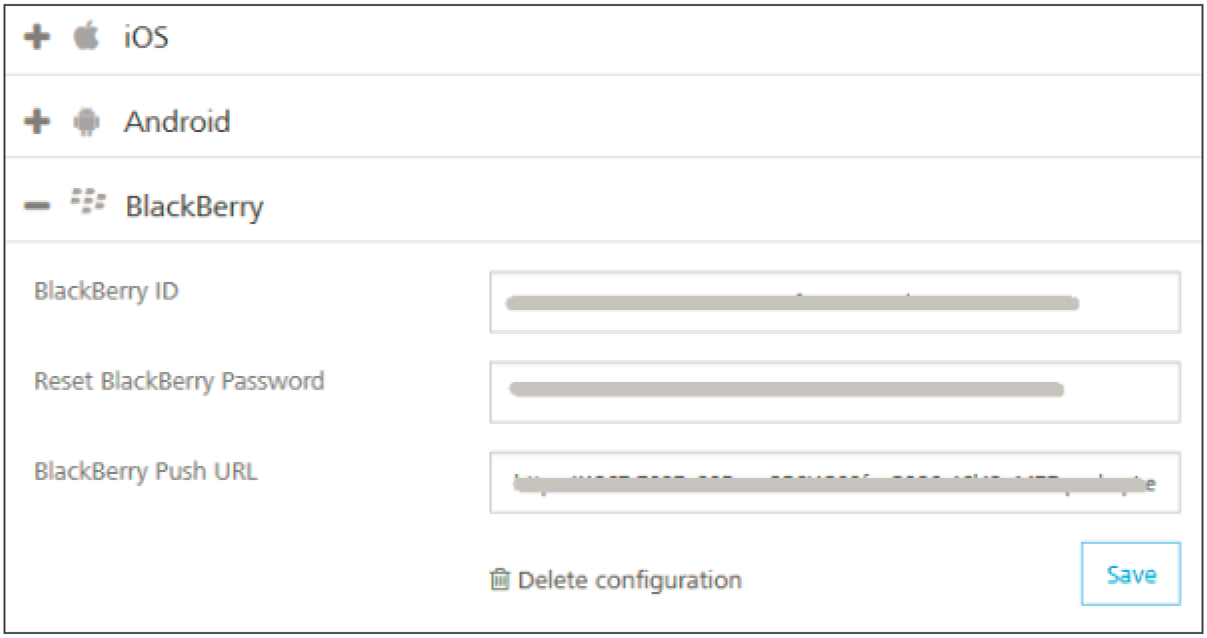
> ***Note:*** Google Cloud Messaging for Android (GCM) is a service that helps you to send data from servers to Android applications on Android devices. This can be a lightweight message telling the Android application that there is new data to be fetched from the server (for example, a movie uploaded by a friend), or it can be a message containing up to 4kb of payload data (so apps like instant messaging can consume the message directly). The GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target device.

9.1.3 BlackBerry

> ***Note:*** Refer to the following section for creating a push certificate:
> http://docs.kony.com/konylibrary/#../Subsystems/KMS_Console_User_Guide/Content/Apps/Adding_Platform.htm

To add BlackBerry Push Certificates for your app, follow these steps:

1. Expand **BlackBerry**. A list of configurable items appear.



BlackBerry ID [redacted]

Reset BlackBerry Password [redacted]

BlackBerry Push URL [redacted]

2. **BlackBerry ID**: Enter the ID.

> *Note:* BlackBerry Identity is a single, master key for BlackBerry products, sites, services, and applications, offering: Simplified access, privacy and security controls, a personalized and customizable experience.

3. **BlackBerry Password**: Enter the password.
4. **BlackBerry Push URL**: Enter the web address.
5. Click **Save** to complete the configuration process.
6. To delete configuration for BlackBerry, click **Delete Configuration**.

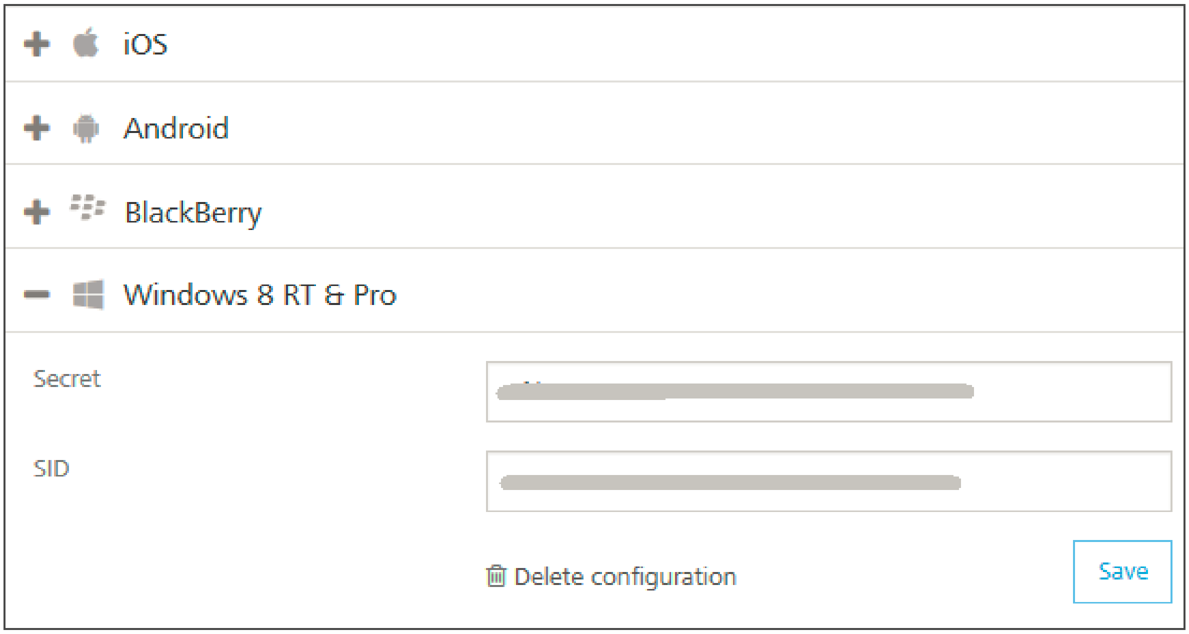
9.1.4 Windows 8 RT & PRO

> ***Note:*** Refer to the following section for creating a push certificate: http://docs.kony.com/konylibrary/#../Subsystems/KMS_Console_User_Guide/Content/Apps/Adding_Platform.htm

> ***Note:*** Windows push certificate is a purchased SSL certificate that is converted to correct format for uploading to MobileFabric.

To add Windows Push Certificates for your app, follow these steps:

1. Expand **WINDOWS 8 RT** and **PRO**. A list of configurable items appear.



Secret [redacted]

SID [redacted]

2. **Secret**: Enter the secret key details.

> *Note:* Windows Secret is an associated secret key that contains strings used in authentication with KMS APIs. It is used in authentication on the client side during registration.

3. **SID**: Enter the SID details, and then click Save to complete the configuration process.

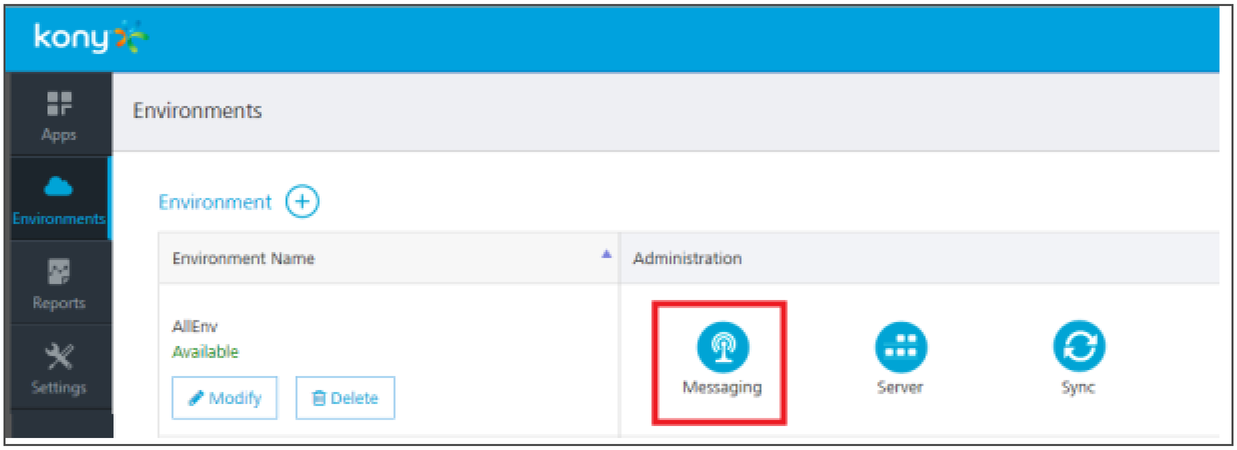
> *Note:* Windows SID is a security identifier that is a unique, immutable identifier of a user, user group or other security principal. A security principal has a single SID for life, and all properties of the principal, including its name, are associated with the SID. This design allows a principal to be renamed (for example, from "John" to "Jane") without affecting the security attributes of objects that refer to the principal.

4. To delete push configuration for Windows, click **Delete Configuration**.

9.2 Accessing Messaging Service Console

The Messaging Service Console allows you to add and manage applications, view the stored certificates, and manage a subscribers list.

You can access Messaging Service console from your MobileFabric cloud account by clicking the **Messaging Service** in the your cloud dashboard.



> ***Note:*** For more information on Messaging Service Console, refer to the following guide: http://docs.kony.com/konylibrary/default.htm#../Subsystems/KMS_Console_User_Guide/Content/Cover_Page.htm?TocPath=Kony Development Cloud|Kony Messaging Services|Kony Messaging Services Console User Guide|_____1.

10. Publishing the App

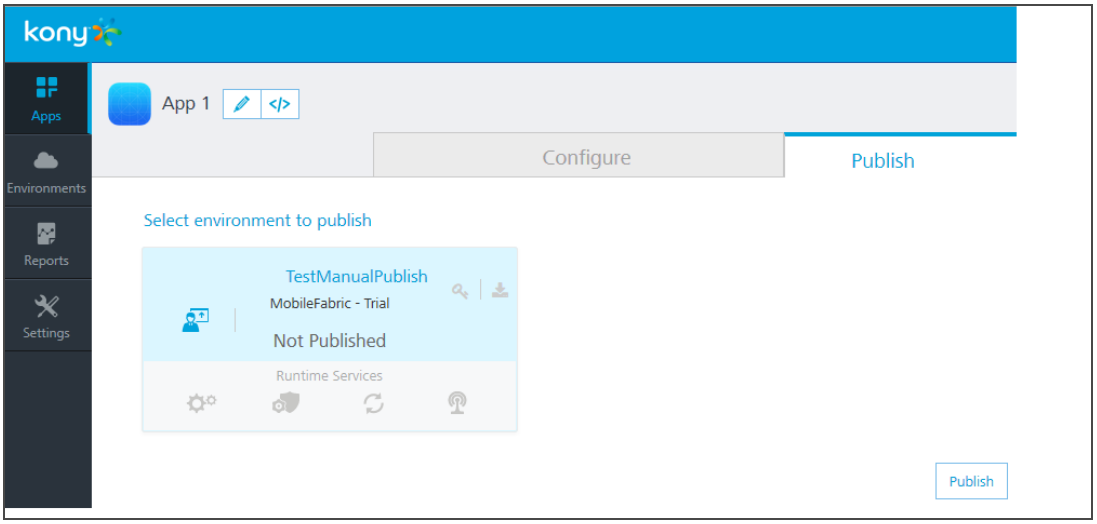
Once you have configured your demo app to use Kony User Repository identity service, you need to publish the app.

Based on environments created, Kony MobileFabric Console allows you to publish apps by using automated publish or manual publish.

- With automated publish, your apps are published to clouds or environments.
- Manual Publish is required only because of some limitations with publishing custom code associated with integration services.

To publish a service, follow these steps:

1. Click the **Publish** tab to view your available environments.
2. Select your target environment, and then click **Publish**.

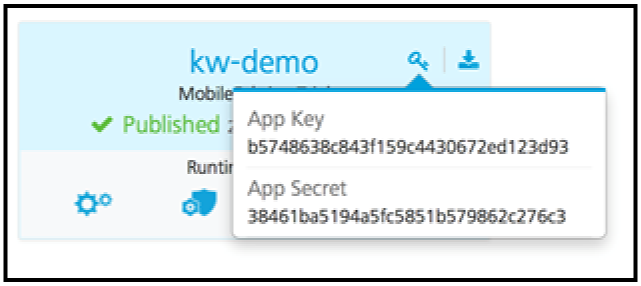


10.1 Testing the Service

Now that the identity service for our app has been published, we can test the service by trying to authenticate as our demo user. To do this, we need the app key, app secret, and the identity service URL of our runtime environment.

For testing the service, follow these steps:

1. From the **Publish** tab, click the key icon of our runtime environment.



App Key
b5748638c843f159c4430672ed123d93

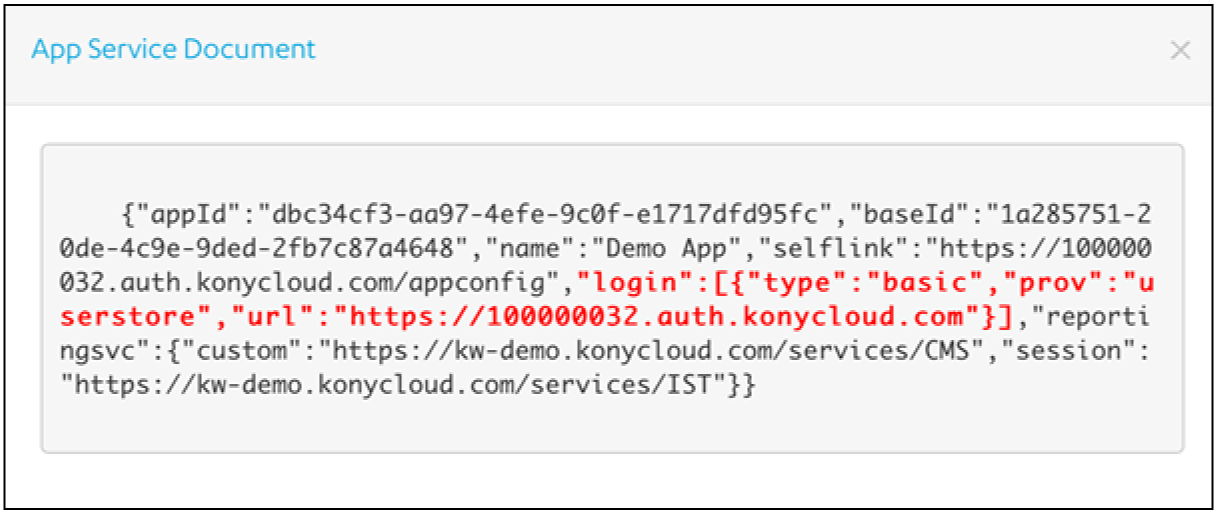
App Secret
38461ba5194a5fc5851b579862c276c3

The app key and app secret are displayed.

The App key, App secret, and Service URL are used to initialize a client app to use MobileFabric services.

> ***Note:*** App Key is unique to each environment.

2. Click the download icon, and then click **App Service Document**.



```
{"appId":"dbc34cf3-aa97-4efe-9c0f-e1717dfd95fc","baseId":"1a285751-20de-4c9e-9ded-2fb7c87a4648","name":"Demo App","selflink":"https://100000032.auth.konycloud.com/appconfig","login":[{"type":"basic","prov":"userstore","url":"https://100000032.auth.konycloud.com"}],"reportingsvc":{"custom":"https://kw-demo.konycloud.com/services/CMS","session":"https://kw-demo.konycloud.com/services/IST"}}
```

This will display a configuration JSON used by the client app to discover what services are available to this app in this environment. The configuration also contains the **login URL** to the user store (highlighted in red).

3. Test the service using the following cURL command for authenticating a user.

> ***Note:*** cURL - a command line tool for getting or sending files using URL syntax. This guide uses the cURL command to represent a mobile device making HTTPS API calls to a Kony MobileFabric environment. cURL is typically pre-installed on Linux and Mac systems. For Windows, go to http://curl.haxx.se/download.html, download cURL, and the SSL libraries required to connect to HTTPS URLs.
>
> For cURL commands and documentation, refer to http://curl.haxx.se/docs/

```
curl -X POST -H "X-Kony-App-Key:
24662f8e60c8a5cd2b2117e37d22fe32" -H "X-Kony-App-Secret:
75aa0fe163ace8f5c4a92245b97d95c6" -H "Accept: application/json" -
H "Content-Type: application/x-www-form-urlencoded" -d
'userid=demo%40kony.com&password=Pass1234'
'https://100000032.auth.konycloud.com/login?provider=userstore'
```

4. The JSON response contains the following elements:

   a. **profile**: Includes user profile information. In this case, it is the user profile attributes from Kony User Repository. If this identity services were connected to an enterprise identity service provider, this would include user profile information from that system.

   b. **provider token**: This is the security token returned from the external identity service provider such as Active Directory or Salesforce. In this case, it is the token returned from the Kony User Repository.

   c. **refresh token**: The refresh token has a longer timeout than the provider token. The refresh token can be used to get a new provider token, but it requires the use of the app key and app secret to request a new provider token.

   d. **claims token**: This is a Kony MobileFabric claims token that will be used for any subsequent calls to Integration, Orchestration, Sync, or Messaging services.

The token values are formatted as JWT tokens and are digitally signed by the server so they can be validated by the server on subsequent calls. They can be decoded using a JWT decoder such as https://developers.google.com/wallet/digital/docs/jwtdecoder to view the data it includes.

The following is a sample JSON response from the cURL command:

```
{
    "profile": {
        "email": "demo@kony.com",
        "userid": "demo@kony.com",
        "firstname": "Demo",
        "lastname": "User"
    },
    "provider_token": {
       "exp": 1412190752000,
       "value":
```

```
"eyAiYWxnIjogIk5PTkUiLCAidHlwIjogImp3cyIgfQ.eyAiX2FjcyI6ICIxMDA
w
MDAwMzIiLCAiZXhwIjogMTQxMjE5MDc1MiwgIl9pc3NtZXRhIjogIi9hcGkvdjE
v
bWV0YWRhdGEvVXl3M0pDdVU4XzVnUEZFNzdCM3ZGdz09IiwgIl92ZXIiOiAidjE
u
MSIsICJfcHJvdl91c2VyaWQiOiAiZGVtb0Brb255LmNvbSIsICJfaWRwIjogInV
z
ZXJzdG9yZSIsICJpc3MiOiAiaHR0cHM6Ly8xMDAwMDAwMzIuYXV0aC5rb255Y2x
v
dWQuY29tIiwgIl9lbWFpbCI6ICJkZW1vQGtvbnkuY29tIiwgImp0aSI6ICJlMDU
w
MmFiNS05OTY4LTQzNzctOWRkNC04ZWI4OWJhMGVhMjEiLCAiaWF0IjogMTQxMjE
4
NzE1MiB9.MCwCFAa5lgjc8PKc3lZwXydRviPtrJXjAhRWSssiS86n4oWyuCZEsc
Z
wkn7aHw",
     "provider": "userstore",
     "params": {}
     },
     "refresh_token":

"eyAiYWxnIjogIk5PTkUiLCAidHlwIjogImp3cyIgfQ.eyAiX3Njb3BlIjogImc
i
LCAiX2FjcyI6ICIxMDAwMDAwMzIiLCAiZXhwIjogMTQxMjI3MzU1MiwgIl9pc3N
t
ZXRhIjogIi9hcGkvdjEvbWV0YWRhdGEvVXl3M0pDdVU4XzVnUEZFNzdCM3ZGdz0
9
IiwgIl92ZXIiOiAidjEuMSIsICJfcHJvdl91c2VyaWQiOiAiZGVtb0Brb255LmN
v
bSIsICJfaWRwIjogInVzZXJzdG9yZSIsICJfYXBwIjogImRlZjljNzM4LTAxNjM
t
```

```
NDgzZS05NzdmLTM1MDIxMjVjMTk4YyIsICJpc3MiOiAiaHR0cHM6Ly8xMDAwMDA
w
MzIuYXV0aC5rb255Y2xvdWQuY29tIiwgIl9lbWFpbCI6ICJkZW1vQGtvbnkuY29
t
IiwgImp0aSI6ICI0MTI1ZmE5Yy1lZDlmLTRjMTItYTYzNC02OGJkOTAwYTNhMTg
i
LCAiaWF0IjogMTQxMjE4NzE1MiwgIl9wdWlkIjogMjgwODI0NjA0OTc5NDU1MDg
2
IH0.MCwCFACJQFUW0C4pYFV2GIvOB0erHrENAhQQ3-Dvfe9ytvCzu-tbJZ_
630lu
XA",
    "claims_token": {
        "value":

"eyAiYWxnIjogIk5PTkUiLCAidHlwIjogImp3cyIgfQ.eyAiX3Njb3BlIjogImc
i
LCAiX2FjcyI6ICIxMDAwMDAwMzIiLCAiX3ZlciI6ICJ2MS4xIiwgIl9pZHAiOiA
i
dXNlcnN0b3JlIiwgIl9hcHAiOiAiZGVmOWM3MzgtMDE2My00ODNlLTk3N2YtMzU
w
MjEyNWMxOThjIiwgImlzcyI6ICJodHRwczovLzEwMDAwMDAzMi5hdXRoLmtvbnl
j
bG91ZC5jb20iLCAiX2VtYWlsIjogImRlbW9Aa29ueS5jb20iLCAiaWF0IjogMTQ
x
MjE4NzE1MiwgImV4cCI6IDE0MTIxOTA3NTIsICJfaXNzbWV0YSI6ICIvYXBpL3Y
x
L21ldGFkYXRhL1V5dzNKQ3VVOF81Z1BGRTc3QjN2Rnc9PSIsICJfcHJvdl91c2V
y
aWQiOiAiZGVtb0Brb255LmNvbSIsICJqdGkiOiAiY2M4MGFkNGEtNGQ0NS00MmF
k
LTk2ZjUtZTY0NzYwZWViZjI2IiwgIl9hdXRoeiI6ICJleUp3WlhKdGFYTnphVzl
1
Y3lJNmUzMHNJbkp2YkdWeklqcGJYWDAiLCAiX3B1aWQiOiAyODA4MjQ2MDQ5Nzk
0
NTUwODYgfQ.MC0CFQCP_1JSQe9stMYjr8P4vrgKYuTn5gIUSx6j_
R9dbjFFCcTCL
AiD6AOdqh0"
,
        "exp": 1412190752000
    }
}
```

11. Settings

Using **Settings**, a superuser can manage tasks such as adding new users, assigning roles to users, deleting users, and configuring reports server.

Settings includes the following sections:

- Users
- Studio
- Reports

11.1 Users

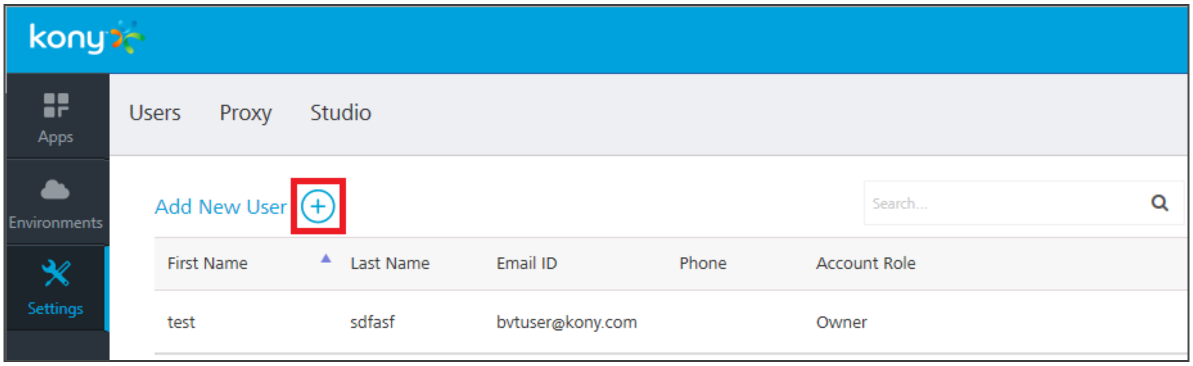
A user is an individual person. Each user needs an account to access the MobileFabric console. A superuser creates user accounts for owners, admins, and members who use the MobileFabric console.

> *Important:* As a user, you must be an admin or owner to access the Users page and perform different tasks based on the role.

11.1.1 How to Add a User

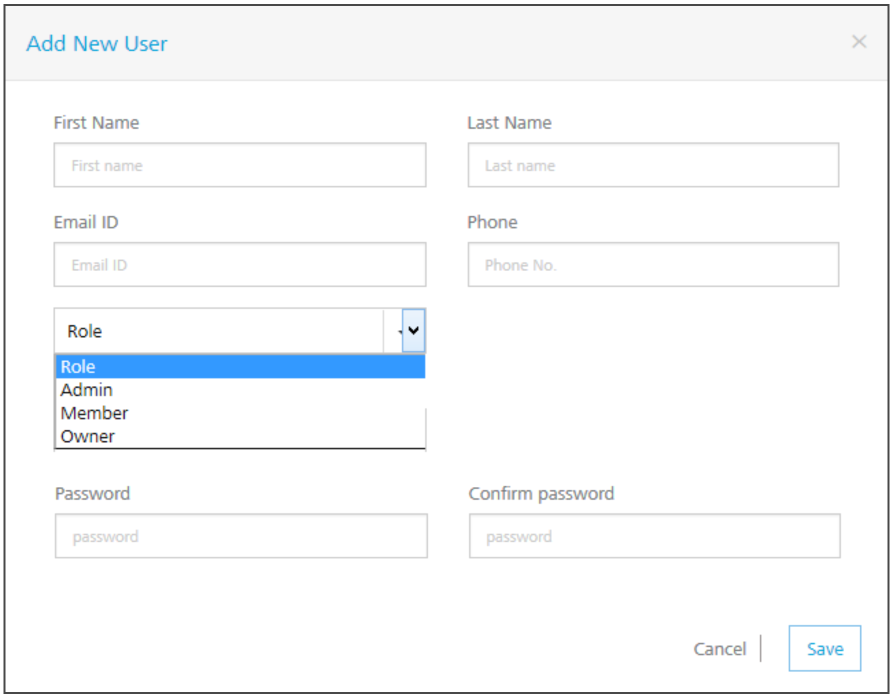
To add a user, follow these steps:

1. In your MobileFabric account, click **Settings**. By default, the **Users** page appears. The **Users** tab is visible to only users who are owners or admins. The page lists all owners, admins, and members of the account.



First Name	Last Name	Email ID	Phone	Account Role
test	sdfasf	bvtuser@kony.com		Owner

2. In the **Users** page, click **Add New User** button. The **Add New User** page appears.



> ***Note:*** All these fields are mandatory except the **Phone** number field.

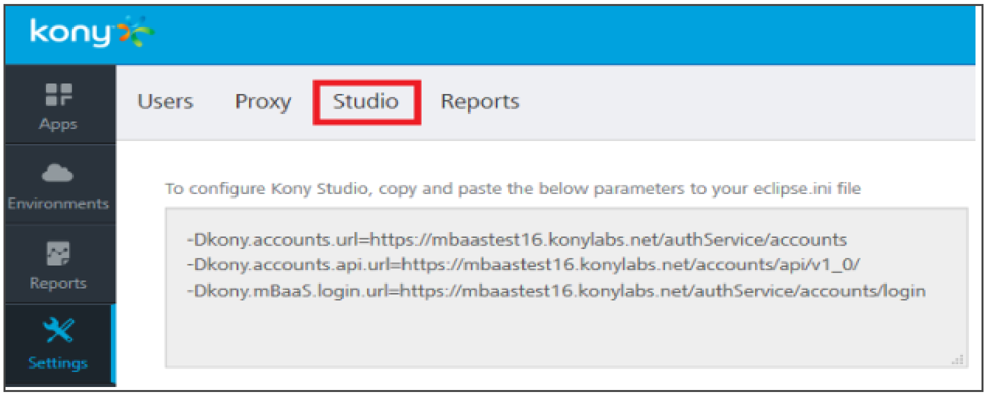
3. Enter the details as required.

4. Click **Save** to save the user details. The system will add the new user in the grid.

11.2 Studio

The Studio tab lists -D parameters that you need to log in to Kony Studio (IDE.) The parameters are generated during MobileFabric installation.

> *Important:* To configure Kony Studio, copy and paste the following parameters in the `the eclipse.ini or eclipse-org.ini` file located in your Kony Studio install folder - for example, `<C:\Program Files\Kony_6.0.3QA\Kony_Studio>eclipse.ini.`



To configure Kony Studio, copy and paste the below parameters to your eclipse.ini file

```
-Dkony.accounts.url=https://mbaastest16.konylabs.net/authService/accounts
-Dkony.accounts.api.url=https://mbaastest16.konylabs.net/accounts/api/v1_0/
-Dkony.mBaaS.login.url=https://mbaastest16.konylabs.net/authService/accounts/login
```

11.2.1 How to Configure -D parameters in Kony Studio in Windows OS

To configure -D parameters in Kony Studio (IDE) in Windows OS, follow these steps:

> *Important:* If you use secured services (HTTPS), configure Java Runtime Environment (JRE) associated with Kony Studio with CA certificates. If your CA certificate is not configured, the login to Kony Studio fails.
>
> If your CA certificate is not configured, the login to Kony Studio fails.
>
> For details, refer to Kony MobileFabric Install Guide > Troubleshoot with SSL Certificate Issues.

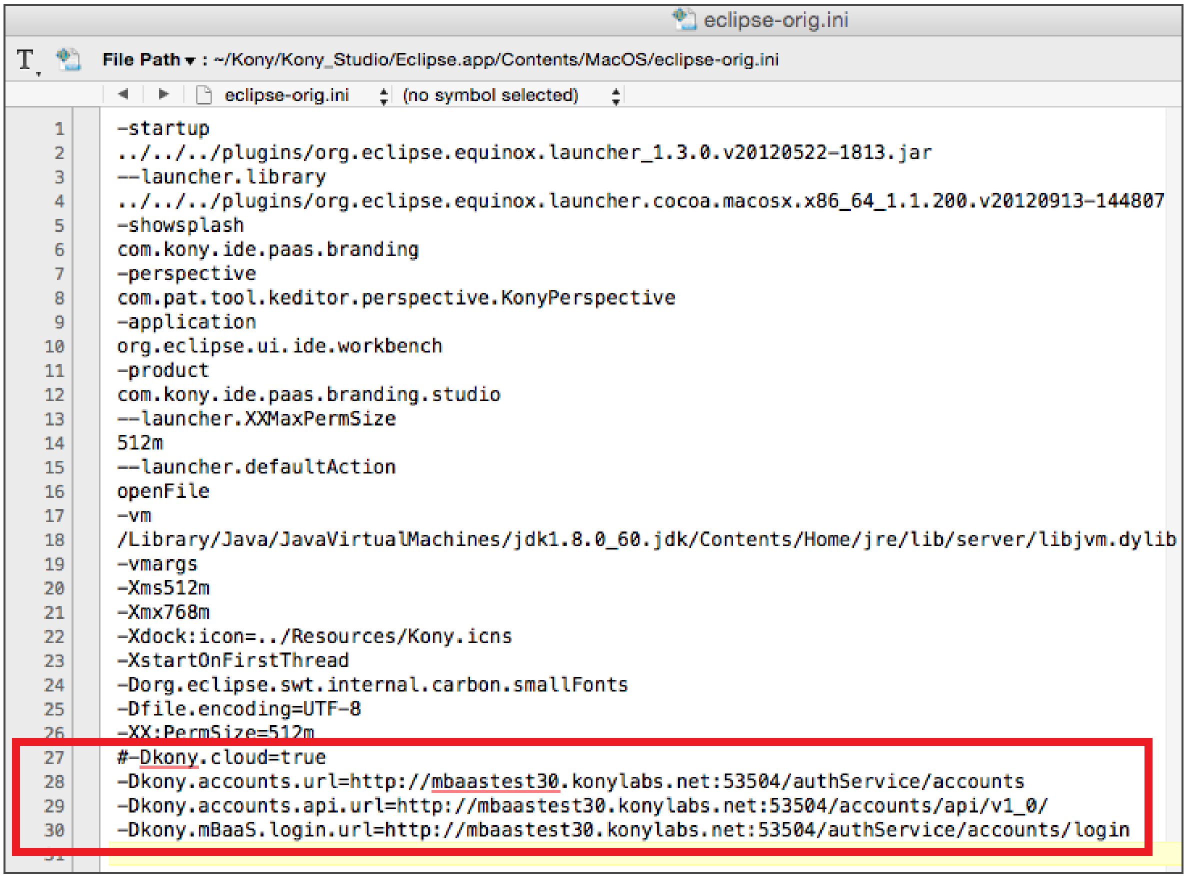
1. Open the `eclipse.ini` file located in your Kony Studio install folder - for example, `<C:\Program Files\Kony_6.0.1GA\Kony_Studio>eclipse.ini`.
2. Copy the **-D parameters** from the **Studio** tab in MobileFabric Console and paste them in the `eclipse.ini` file.
3. Save the `eclipse.ini` file and restart Kony Studio.

11.2.2 How to Configure -D parameters in Kony Studio in Mac OS

To configure -D parameters in Kony Studio (IDE) in Mac OS, follow these steps:

1. Open the `eclipse-org.ini` file located in your Kony Studio install folder - for example, `<Install Drive>:\Kony_Studio>eclipse-org.ini`.
2. Copy the **-D parameters** from the **Studio** tab in MobileFabric Console and paste them in the `eclipse-org.ini` file.
3. In the `eclipse-org.ini` file, comment the line for `-Dkony.cloud=true` parameter by adding an hash at the beginning of the line. For example, `#-Dkony.cloud=true`.

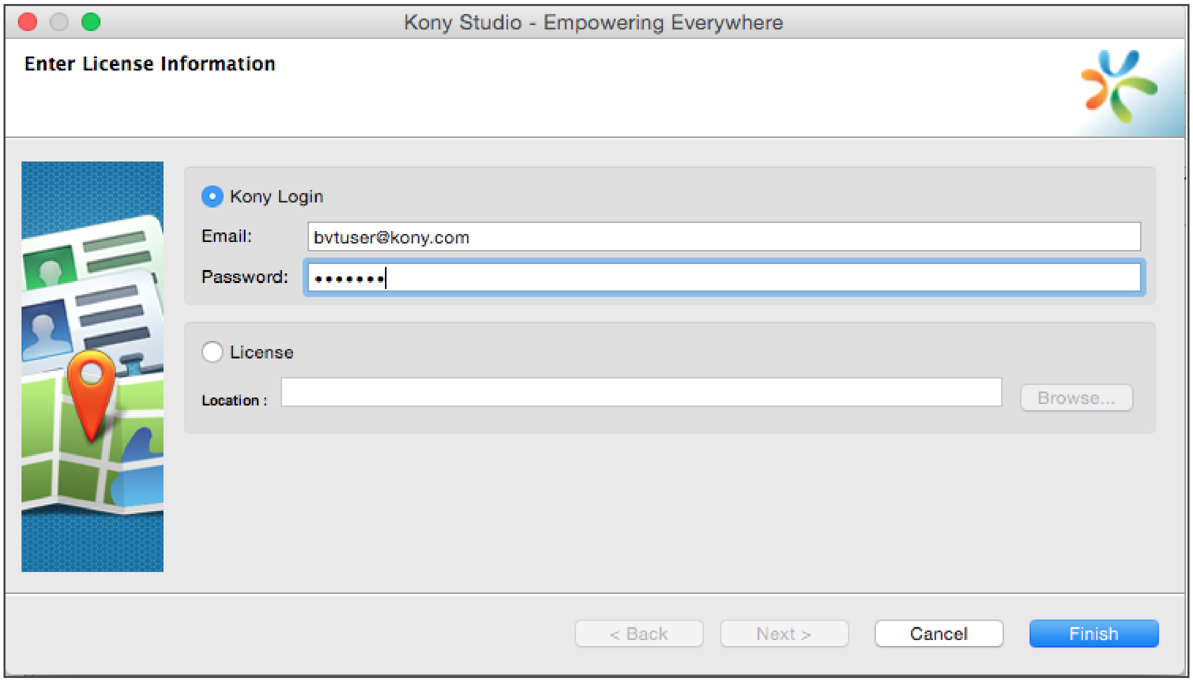
   Ensure that no spaces at the end of -D parameters lines that you added in this file.


eclipse-orig.ini
File Path ▾ : ~/Kony/Kony_Studio/Eclipse.app/Contents/MacOS/eclipse-orig.ini
eclipse-orig.ini
(no symbol selected)

```
-startup
../../../plugins/org.eclipse.equinox.launcher_1.3.0.v20120522-1813.jar
--launcher.library
../../../plugins/org.eclipse.equinox.launcher.cocoa.macosx.x86_64_1.1.200.v20120913-144807
-showsplash
com.kony.ide.paas.branding
-perspective
com.pat.tool.keditor.perspective.KonyPerspective
-application
org.eclipse.ui.ide.workbench
-product
com.kony.ide.paas.branding.studio
--launcher.XXMaxPermSize
512m
--launcher.defaultAction
openFile
-vm
/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/jre/lib/server/libjvm.dylib
-vmargs
-Xms512m
-Xmx768m
-Xdock:icon=../Resources/Kony.icns
-XstartOnFirstThread
-Dorg.eclipse.swt.internal.carbon.smallFonts
-Dfile.encoding=UTF-8
-XX:PermSize=512m
#-Dkony.cloud=true
-Dkony.accounts.url=http://mbaastest30.konylabs.net:53504/authService/accounts
-Dkony.accounts.api.url=http://mbaastest30.konylabs.net:53504/accounts/api/v1_0/
-Dkony.mBaaS.login.url=http://mbaastest30.konylabs.net:53504/authService/accounts/login
```

4. Save the `eclipse-org.ini` file and restart Kony Studio.

5. By default, the system asks you to log in to Kony Studio. Enter your MobileFabric login details. For example, `admin@localhost/MF-iXXXXX`.



Kony Studio - Empowering Everywhere

Enter License Information

● Kony Login

Email: bvtuser@kony.com

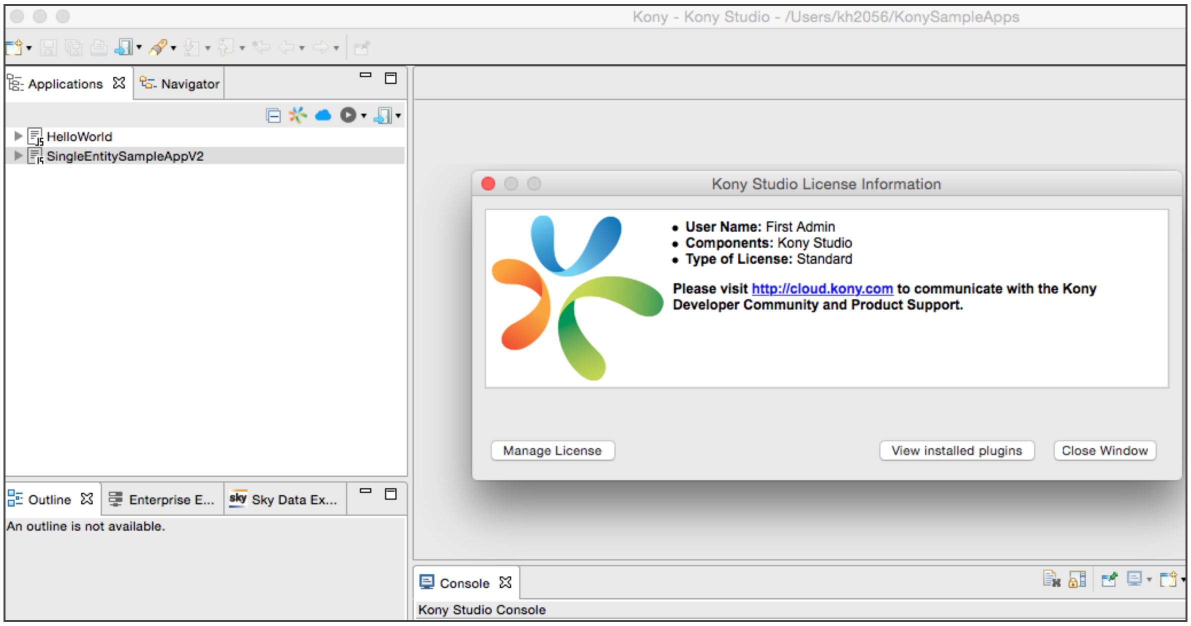
Password: •••••••

○ License

Location :

6. If the system does not prompt you to log in to Kony Studio by default, navigate to **Help** > **About Kony License**. The **Kony Studio License Information** dialog appears.

7. Click **Manage License** and enter log in details.



Kony - Kony Studio - /Users/kh2056/KonySampleApps

Kony Studio License Information

- **User Name:** First Admin
- **Components:** Kony Studio
- **Type of License:** Standard

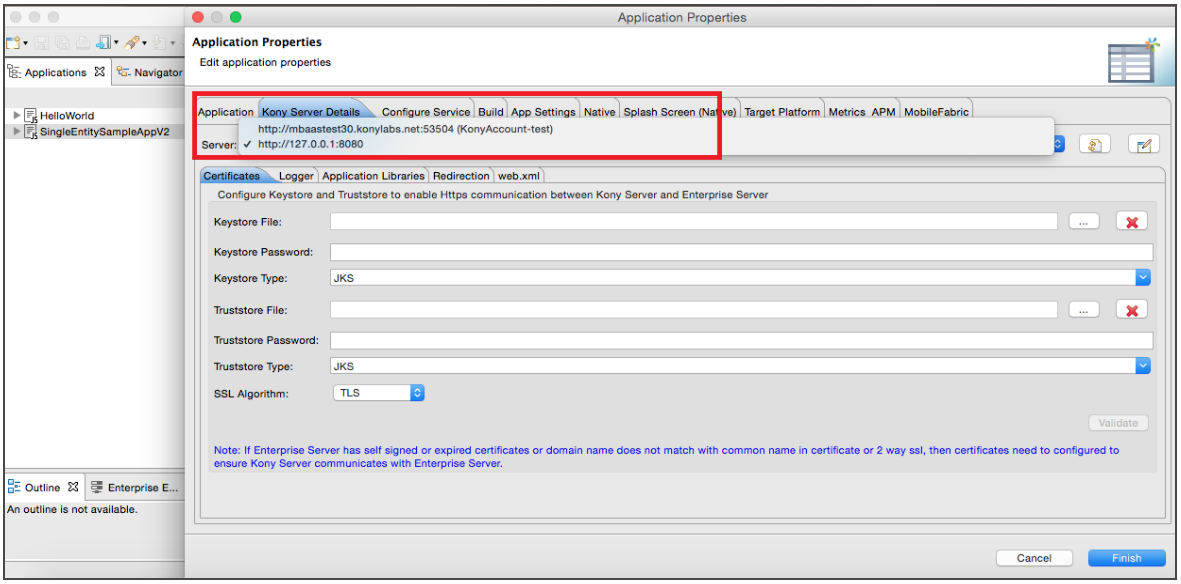
Please visit http://cloud.kony.com to communicate with the Kony Developer Community and Product Support.

An outline is not available.

Kony Studio Console

8. In **Applications** view, right-click your **Project**, and click **Properties**. The **Application Properties** dialog appears.

9. Click the **Kony Server Details** tab to view your MobileFabric Server is listed.



Application Properties
Edit application properties

Server: http://mbaastest30.konylabs.net:53504 (KonyAccount-test)
✓ http://127.0.0.1:8080

Configure Keystore and Truststore to enable Https communication between Kony Server and Enterprise Server

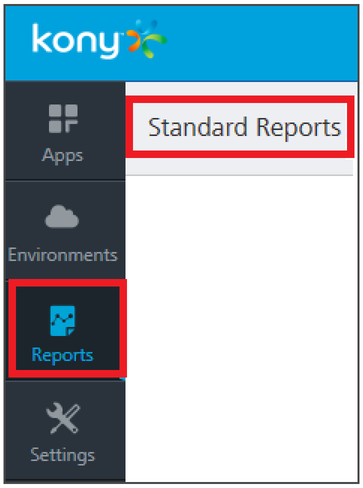
Keystore Type: JKS
Truststore Type: JKS
SSL Algorithm: TLS

Note: If Enterprise Server has self signed or expired certificates or domain name does not match with common name in certificate or 2 way ssl, then certificates need to configured to ensure Kony Server communicates with Enterprise Server.

An outline is not available.

11.3 Reports

Under the **Settings**, the **Reports** tab allows you to configure the JasperReports Server. Once you complete JasperReports Server configuration, the **Reports** page (shown below) displays data (reports) from the JasperReports Server.



Currently, the **Reports** page displays only **Standard Reports**. To view standard reports, click the report. For more details on standard reports, refer to Kony Reporting and Analytics - Standard Reports.

11.3.1 How to Configure JasperReports Server

> ***Important:*** Before configuring the JasperReports Server in the Reports tab, ensure that you have installed the JasperReports Server and configured MobileFabric Console in the JasperReports Server. For more details about how to set up the JasperReports Server, refer to Kony Analytics and Reporting Installation Guide.

1. In your MobileFabric account, click **Settings** > **Reports**. The **Reports** page appears.
2. In the **Jasper URL** text box, enter the JasperReports Server URL.
3. In the **Username** text box, type jasperadmin.

   > ***Note:*** Enter credentials for jasperadmin. The default credentials for jasperadmin:
   > username = jasperadmin
   > password = jasperadmin

4. In the **Password** box, type `jasperadmin`.
5. Click **Save** to the save the JasperReports Server. The confirmation message appears.

   After you configured JasperReports Server successfully, you can access the standard reports from **Reports** > **Standard Reports** page.

12. News and Weather Application

This section discusses the details of building and configuration of a sample News and Weather application by using Kony MobileFabric. Following details are included in this section for the News and Weather application:

- Overview of the News and Weather Application
- Creating News and Weather Application in MobileFabric
- Creating the Identity Service for News and Weather App
- Creating the Integration Service for News and Weather App
- Creating the Orchestration Service for News and Weather App

12.1 Overview of the News and Weather Application

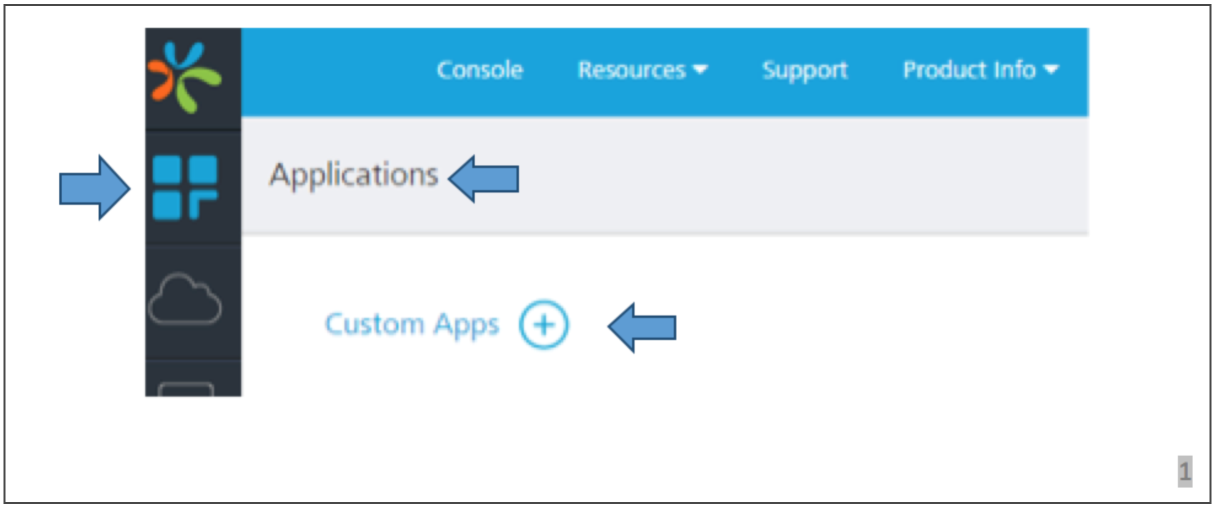
This document describes how to create a Kony MobileFabric Integration Services and Orchestration Services to get News (by category, Local News) and Weather Forecast from different Web services.

12.2 Creating News and Weather Application in MobileFabric

Follow the steps below to create a MobileFabric Application using the account that has been created using the Kony Cloud Portal.

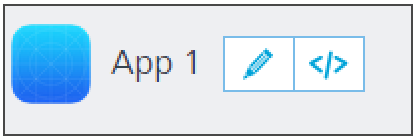
1. Log in to your Kony Cloud Portal.
2. Click **Apps** in the left menu.

3. Under the Applications, click the **Custom Apps Plus (+) button** to create a new application.



A new Kony MobileFabric Application is created, and you will be navigated to the app configuration page.

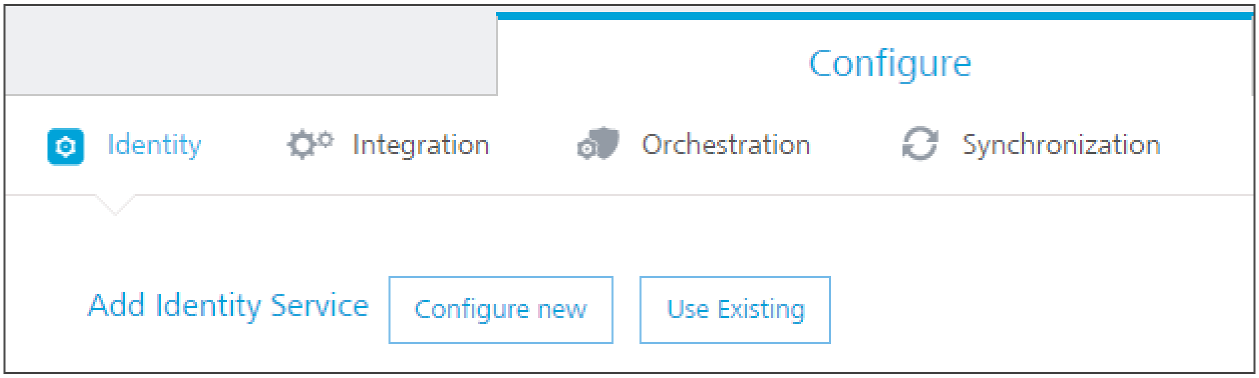
4. To change the app name, click on the pencil (edit) icon next to the app title, shown below:



12.3 Creating the Identity Service for News and Weather App

An identity is a service that validates the authentication of your users before they can access your application. In this application we are not configuring any identity services so that any user can access the application.

You can configure an identity service under the **Identity** tab, shown below:



The following sections helps you to create a MobileFabric Application using the account that is created using the Kony Cloud Portal.

12.4 Creating the Integration Service for News and Weather App

The Integration tab is where you define services for your application.

A service is an application component that represents the application interaction with the external data source. A service definition comprises the meta-data or the configurations required to exchange data with the external data source. For example, the configurations can be service type, service ID, input parameters, output parameters, preprocessors and postprocessors, target URL, authentication credentials if required, and type (HTTP/HTTPS).

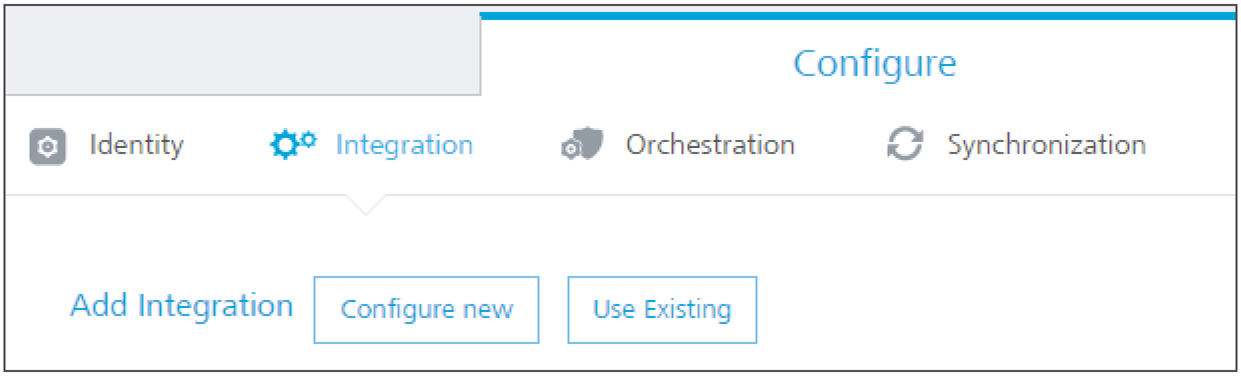
The service definition enables the application to exchange data with any external data source. The MobileFabric provides back-end for connecting to a Web service and an XML service. Even if the external data source does not expose the services to these well-known interfaces, the developer can build a Java service.

Let us create three different services for our application:

- News Service
- Weather Service
- Location Lookup Service

12.4.1 How to Create and Configure a News Service

We need to configure these services in **Integration** tab, shown below:



1. Click **Configure New**.

2. In the **Service Name** text box, provide a name for the integration service - for example, News.

3. From the **Service Type** dropdown, select XML.

4. In the **Base URL** box, type https://news.google.com/news/section?output=rss&.

5. Click **Save and Continue** to add the operations.

6. In **Operation name** field, type GetNewsForCategory, and click **Configure Operation**. You will be redirected to operation details page where you can configure the operation.

To configure the GetNewsForCategory operation, follow these steps:

1. In the **New Operation Path** field, add the value `topic=$category`.

2. From the **HTTP Methods** drop-down, select **GET** method for this operation.

3. Now Select "Public" in Operation security Level dropdown. We are making this service public which means we don't any authentication to access this operation

4. Click on **Input** tab. In this section we will configure the input parameters for the operation.

5. Add category as input field and test value `w` which will retrieve the world news. Click the **Test** button, which is at bottom of the page. You will see the <testdata></testdata>in **Result** section, which is on right pane. The result will be converted to a JSON before being sent to the device.

6. Now click **Response** tab that will show the XML response from the service. To send the response to the client we have to configure the output tab.

7. Click the **Output** tab. Click the **Plus(+)** button to insert the output fields.

   The following is a list of output fields you have to create. These fields will be sent to the mobile client.

ID	Path	Scope	Datatype	Collection ID	Record ID	Format	Format Value
news_list	Channel	Response	Collection			None	
news_item	Item	Response	Record	news_list		None	
Title	Title	Response	String		news_item	None	
Link	link	Response	String		news_item	None	
Description	description	Response	String		news_item	None	

8. Once you create above all fields, click the **Test** button to see the **Result** tab.

   Now you will see all the data, which will be sent to mobile client.

9. Click **Save Operation**.

To create one more operation for LocalNews, follow these steps:

1. In the **New Operation Path** field, add the value `q=$cityName`.
2. From the **HTTP Methods** drop-down, select **GET**method for this operation.
3. Now Select “Public” in Operation security Level dropdown. We are making this service public which means we don’t any authentication to access this operation
4. Click on **Input** tab. In this section we will configure the input parameters for the operation.
5. Add `cityName` as input field and test value `New York` which will retrieve the local news. Click the **Test** button, which is at bottom of the page. You will see the <testdata></testdata>in **Result** section, which is on right pane. The result will be converted to a JSON before being sent to the device.
6. Now click **Response** tab that will show the XML response from the service. To send the response to the client we have to configure the output tab. The result will be converted to a JSON before being sent to the device.
7. Click the **Output** tab. Click the **Plus(+)** button to insert the output fields.

   The following is a list of output fields you have to create. These fields will be sent to the mobile client.

ID	Path	Scope	Datatype	Collection ID	Record ID	Format	Format Value
news_list	Channel	Response	Collection			None	
news_item	Item	Response	Record	news_list		None	
Title	Title	Response	String		news_item	None	
Link	Link	Response	String		news_item	None	
Description	description	Response	String		news_item	None	

8. Once you create above all fields, click the **Test** button to see the **Result** tab.

   Now you will see all the data, which will be sent to mobile client.

9. Click **Save Operation**.

12.4.2 How to Create and Configure a Weather Service

1. In the **Integration** page, click **Configure New**.

2. In the **Service Name** text box, provide a name for the integration service - for example, Weather.

3. From the **Service Type** dropdown, select SOAP.

4. Enter the following URLs to complete the service definition:

   - In the **Base URL** box, type: http://wsf.cdyne.com/WeatherWS/Weather.asmx

   - In the **WSDL URL**box, type: http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl

5. Click **Save and Continue** to retrieve the WSDL. Each of the available operations are listed in the **Operation name** drop-down list. Select **GetCityForecastByZip** and **GetCityWeatherByZip** check boxes from the drop-down.

5. Click **Add Operation**. This will create two operations under your Weather service that maps to the SOAP web service methods.

6. To test and edit the GetCityWeatherByZip, click the **Settings** button and choose **Edit**.

   The operation details window appears. A sample Web service request is provided showing a placeholder for any input parameters. For the GetCityWeatherByZip, the ZIP is the only input parameter displayed as `<ns1:ZIP>?XXX?</ns1:ZIP>` in the sample request. At this point, we could hard code a value, but since we want our app to provide the zip code, we need to provide an input variable name: `<ns1:ZIP>$zip</ns1:ZIP>`. We then need to define that variable under the Input tab including a test value of 10036.

7. On the **Output** tab, enter the following parameters and path.

   - city //City

   - state //State

- temp //Temperature

9. Click the **Test** button again and the result will be displayed as XML. The result will be converted to a JSON before being sent to the device.

10. Edit the GetCityForecastByZip in the same way. This service returns a repeating data structure for each day providing that day's weather forecast. This requires the use of the collection ID under the output tab to create a repeating set of JSON objects. After creating the ZIP input parameter the same way as the previous service, enter the following output parameters:

ID	Path	Datatype	Collection ID
ForecastList	//ForecastResult/Forecast	Collection	
Date	Forecast/Date	Record	ForecastList
Desc	Forecast/Description	String	ForecastList
Low	Forecast/Description	String	ForecastList
High	Forecast/Temperatures/MorningLow	String	ForecastList
Daypct	Forecast/Temperatures/DaytimeHigh	String	ForecastList
Nightpct	Forecast/ProbabilityOfPrecipiation/Nighttime	String	ForecastList

Test your service and you will see the resulting XML showing the repeating collections of forecasts.

12.4.3 How to Create and Configure a Location Lookup Service

1. In the **Integration** page, click **Configure New**.
2. In the **Service Name** text box, provide a name for the integration service - for example, LocationLookup.
3. From the **Service Type** drop-down, select XML.
4. In the **Base URL** box, type http://nominatim.openstreetmap.org/reverse?format=xml.
5. Click **Save and Continue** to add the operations.
6. In **Operation name** field, type ReverseGeoCode, and click **Configure Operation**. You will be redirected to operation details page where you can configure the operation.

To configure the ReverseGeoCode operation, follow these steps:

1. In the **New Operation Path** field, add the value `&lat=$lat&lon=$lon`.
2. From the **HTTP Methods** drop-down, select **GET**method for this operation.
3. Now Select "Public" in Operation security Level dropdown. We are making this service public which means we don't any authentication to access this operation
4. Click on **Input** tab. In this section we will configure the input parameters for the operation.
5. Add **lat** as input field and test value `40.7127`. Add **lon** as input field and test value `-74.0059`, which will retrieve the location details. Click the **Test** button, which is at bottom of the page. You will see the <testdata></testdata>in **Result** section, which is on right pane. The result will be converted to a JSON before being sent to the device.
6. Now click **Response** tab that will show the XML response from the service. To send the response to the client we have to configure the **Output** tab.

7. Click the **Output** tab. Click the **Plus(+)** button to insert the output fields.

   The following is a list of output fields you have to create. These fields will be sent to the mobile client.

ID	Path	Datatype	Collection ID	Record ID
zip	//addressparts/postcode	String		
city	//addressparts/city	String		

12.5 Creating the Orchestration Service for News and Weather App

The Integration tab is where you define services for your application.

Orchestration is the coordination or integration of several services and exposing them as a single service. The mix of services supports the automation of business processes.

The following types of orchestration services are available in Kony MobileFabric:

- **Composite Services**: Allows you to run two or more services either concurrently or sequentially.
- **Looping Services**: Allows you to run a single service in a loop till the loop ends or exit criteria is met.

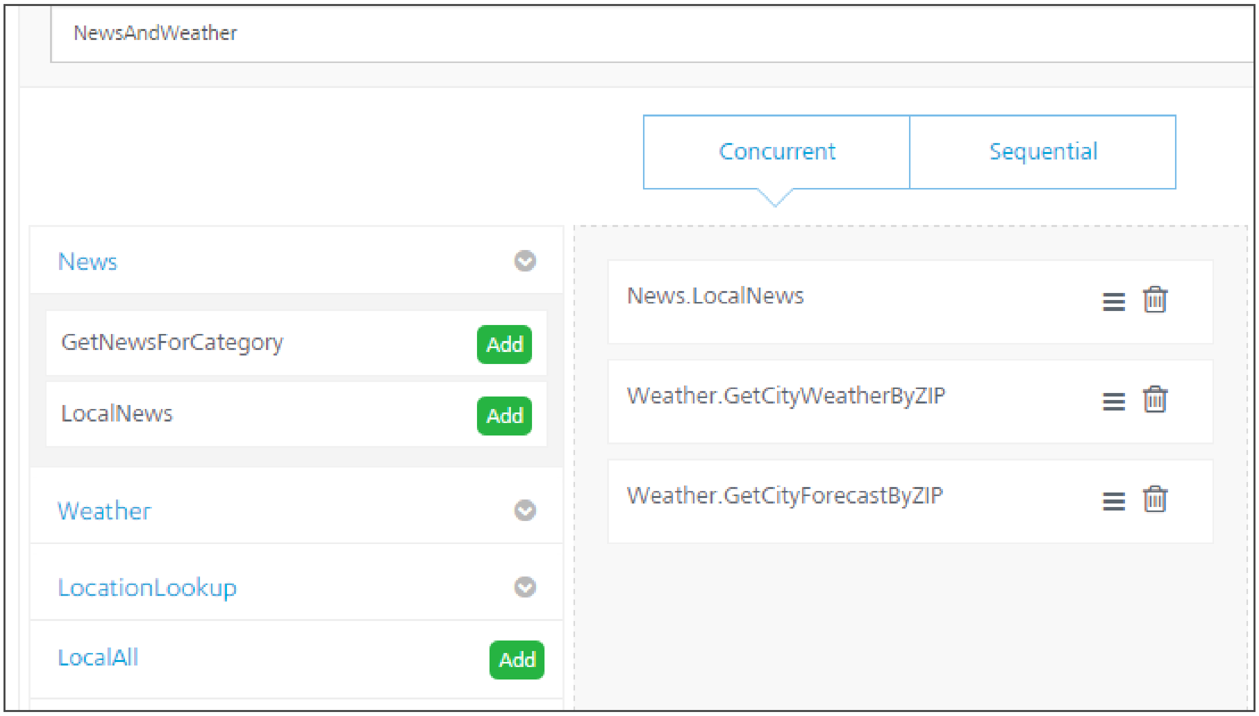
We will create the two orchestration services:

- NewsAndWeather - Concurrent Service
- LocalAll - Sequential Service

12.5.1 Creating News and Weather Orchestration Service

We need to configure these services in **Orchestration** tab, shown below:

1. Click **Create Composite**.
2. In the service name, enter **NewsandWeather** and click the **Concurrent** tab.
3. Now add the below integration services, by clicking the **Add** button next to each service, shown below:
   - LocalNews
   - GetCityWeatherByZip
   - GetCityForecastByZip



NewsAndWeather

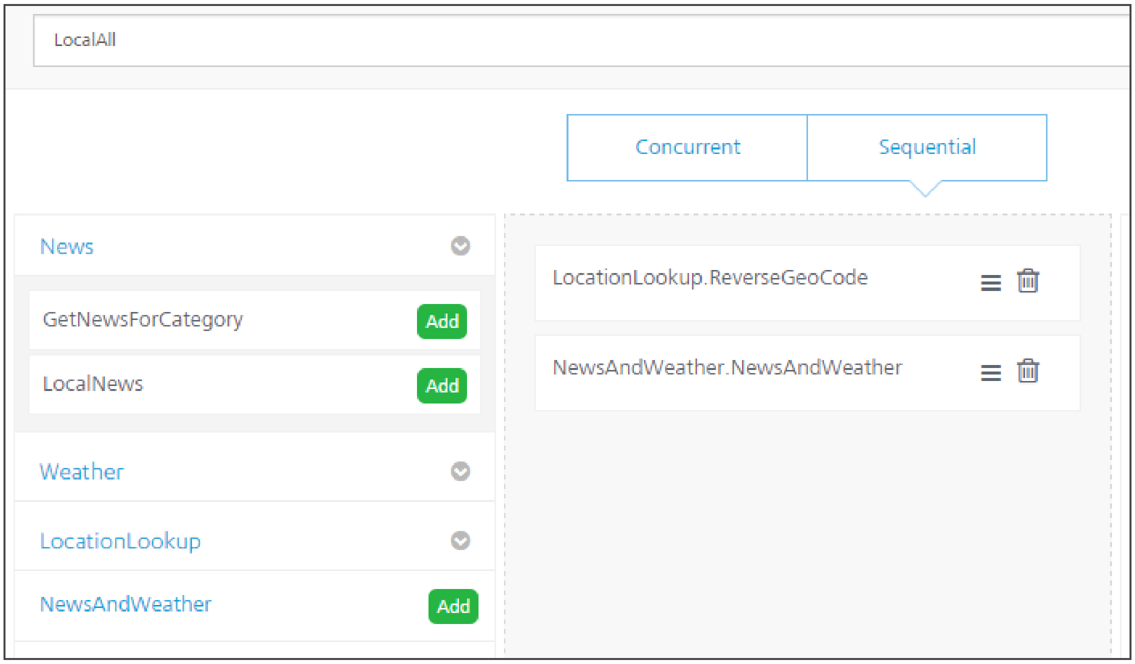
4. Click the **Save** button.

12.5.2 Creating LocalAll Orchestration Service

1. In the **Orchestration** page, click **Create Composite**.
2. In the service name, enter **LocalAll** and click the **Sequential** tab.

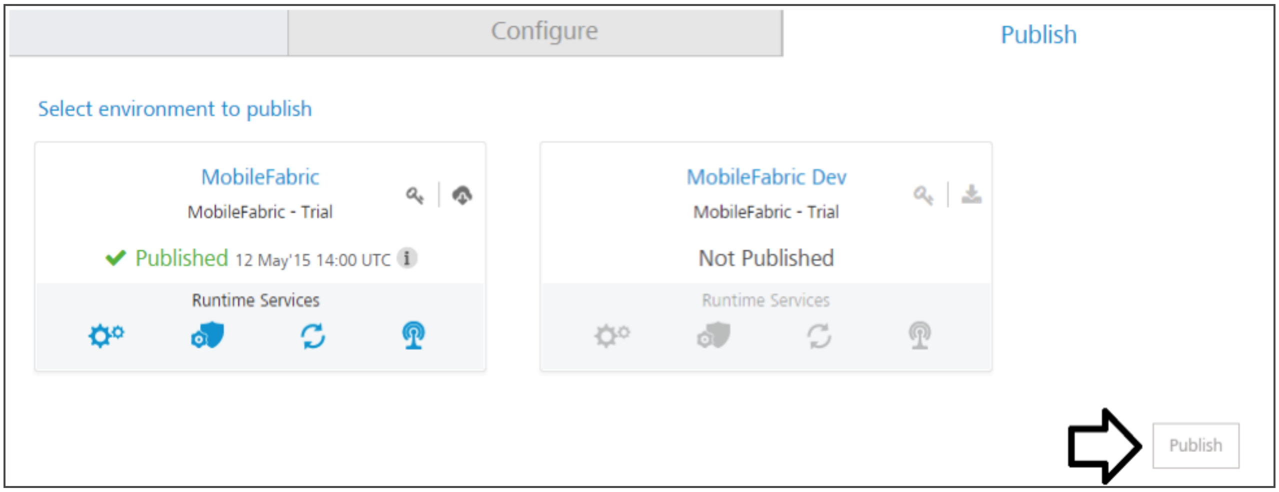
3. Now add the below integration services, by clicking the **Add** button next to each service, shown below:

    - ReverseGeoCode
    - NewsandWeather that you have created in the Creating News and Weather Orchestration Service section.



4. Click the **Save** button.

5. Now click the **Publish** tab (see below) to display the **Publish** page.



6. Select the environment and click the **Publish** button to publish the app into cloud.