



Pure commitment.

Configuring IPsec VPN in Cloud Native Infrastructure



UKC-GEN-476

Introduction

This blueprint provides instructions for deploying an instance into your OpenStack project to function as a VPN endpoint, enabling secure connection:

- To other projects
- To vCloud Director VDCs
- Back to your own infrastructure in-house

This blueprint uses a pfSense virtual firewall appliance to provide the IPsec VPN endpoint. UKCloud provides a Cloud Native Infrastructure image for the pfSense appliance in the public catalog, however, you can upload your own image if you prefer.

You can find the HEAT templates and other files used in this blueprint in the UKCloud GitHub repository at:

<https://github.com/UKCloud/vpn-on-openstack>

The repository contains the following:

- `pfsense.yaml` — A HEAT template to deploy all the resources needed for an IPsec VPN endpoint
- `properties.yaml` — The HEAT parameters used with the `pfsense.yaml` template
- `infrastructure.yaml` — A sample HEAT template to deploy all the prerequisite infrastructure used by the `pfsense.yaml` template
- `vshield-configuration.yaml` — A sample configuration to set up a vCNS Edge (previously called vShield Edge) as an IPsec endpoint on a VDC

IN THIS BLUEPRINT ►

Network infrastructure requirements	3
Set up the VPN stack	5
Create the VPN tunnel	8
Configure the second endpoint	13
Check the tunnel status	14
About UKCloud	15

Network infrastructure requirements

The `pfsense.yaml` HEAT template used by this blueprint assumes that you've already started deploying network resources and instances in your OpenStack project, and that you now want to set up an IPsec VPN tunnel to connect your project to external resources.

It assumes that the following resources already exist:

- A network and subnet on which you are creating instances
- A router to connect your subnet to your internet-facing external network
- An SSH keypair used to create instances
- A jumpbox or bastion Linux server instance created on your subnet
- A floating IP address allocated and assigned to your jumpbox instance
- A security group allowing inbound SSH access to your jumpbox instance

If you have not already created these resources, we've provided a sample HEAT template (`infrastructure.yaml`) in the GitHub repository to get you started. This template creates the resources listed above for you, including an internal network with the subnet `192.168.1.0/24` that uses:

- an allocation pool of `192.168.1.20-192.168.1.200`
- Google's public DNS server, `8.8.8.8`

To deploy the `infrastructure.yaml` template, pass the following URL directly into the OpenStack Horizon UI:

```
https://raw.githubusercontent.com/UKCloud/vpn-on-openstack/master/infrastructure.yaml
```

Select Template ✕

Template Source *

URL

Template URL ⓘ

Environment Source

File

Environment File ⓘ

No file selected.

Description:
Use one of the available template source options to specify the template to be used in creating this stack.

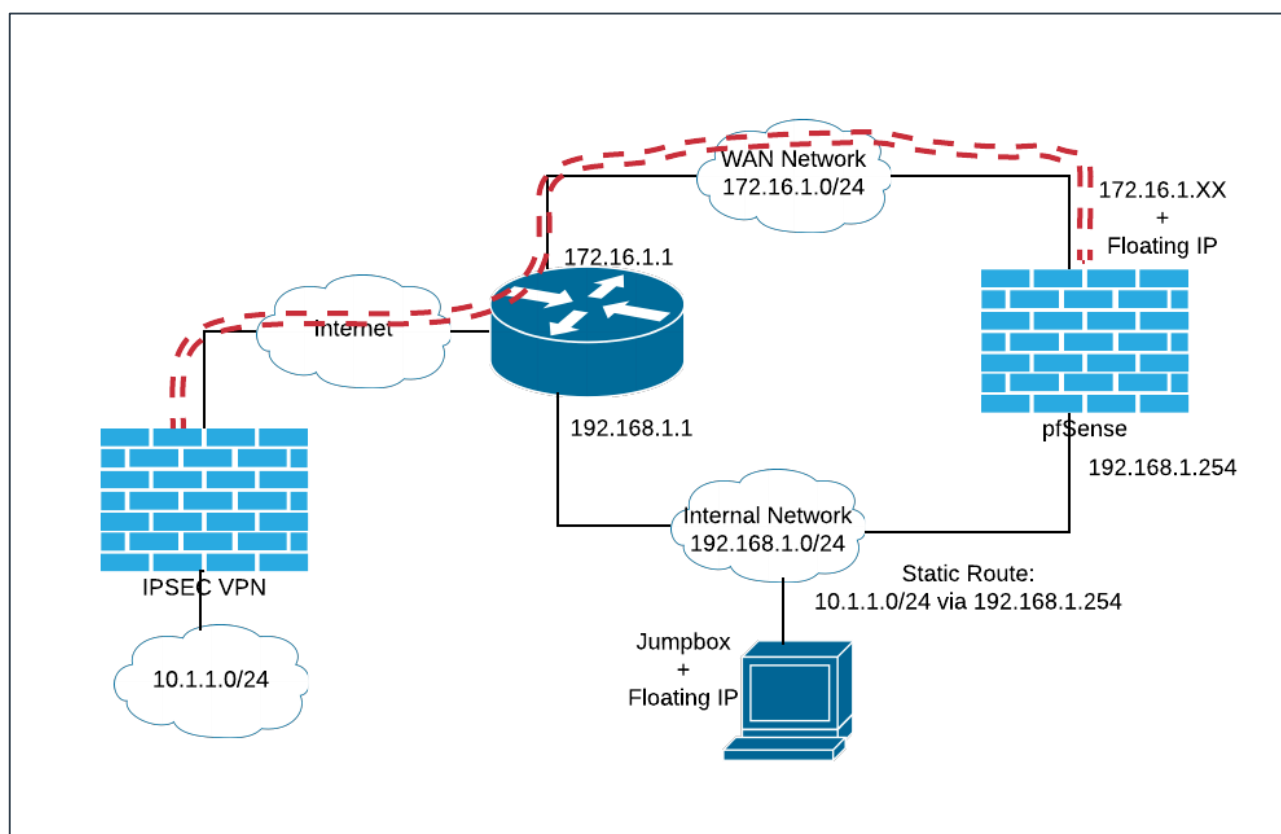
Alternatively, you can clone the repository locally and use the OpenStack CLI to create the stack with the following command:

```
openstack stack create --parameter flavor=t1.small --parameter image="CentOS 7" --wait -t infrastructure.yaml Infrastructure
```

Set up the VPN stack

The pfSense appliance assumes that it will have at least two network interface controllers (NICs) attached to it: one attached to the WAN and one or more attached to internal networks. To fit this into an existing infrastructure deployed in your project, and to avoid any nasty routing loops, the `pfsense.yaml` HEAT template creates a new WAN network along with an interface on your existing router. This provides the WAN interface for pfSense and has a floating IP address associated with it for the internet-facing endpoint of the VPN tunnel.

The LAN interface of the pfSense appliance attaches to your existing internal network. Both WAN and LAN interfaces in the pfSense appliance image in the public catalog are configured for DHCP. This enables the interface to seamlessly configure against your existing infrastructure.



Preparing your configuration

Use the `pfsense.yaml` and `properties.yaml` files provided in the UKCloud GitHub repository to prepare your infrastructure:

1. If you haven't already done so, clone or take a copy of the repository on the UKCloud GitHub:

<https://github.com/UKCloud/vpn-on-openstack>

2. Update the `properties.yaml` file to suit your deployed infrastructure.

For example, if you used the `infrastructure.yaml` HEAT template provided to build your core infrastructure, the `properties` should look similar to the following:

```
parameters:
  key_name: jumpbox_key
  pfsense_flavor: t1.tiny
  router: InternetGW
  internal_network: Internal
  internal_address: 192.168.1.254
  internal_net_cidr: 192.168.1.0/24
  remote_net_cidr: 10.1.1.0/24
```

where:

- `InternetGW` is the name of your router
- `Internal` is the name of your existing network
- `internal_address` is the address on your existing network that will be assigned to the LAN interface on the appliance

Note! The `pfsense.yaml` HEAT template assumes that your VPN tunnel is connecting to only a single remote network. If you need to route to multiple remote networks, after completing the deployment, you'll need to follow a few more configuration steps. For more information, ask to speak to one of our DevOps consultants or Cloud Architects.

Launching the stack

Use the OpenStack CLI tools to launch the stack by running the following command:

```
openstack stack create --enable-rollback -t pfsense.yaml -e properties.yaml --wait
VPNStack
```

You can also launch the stack through the Horizon Web UI by selecting the `pfsense.yaml` and `properties.yaml` files appropriately when prompted:

Select Template ✕

Template Source *

URL

Template URL ⓘ

Environment Source

File

Environment File ⓘ

properties.yaml

Description:

Use one of the available template source options to specify the template to be used in creating this stack.

Create the VPN tunnel

Connect to the pfSense UI

When the HEAT template has finished deploying your VPN stack, connect to the pfSense appliance to configure the IPsec tunnel. You can perform some configuration steps through the OpenStack console, or an SSH connection to the appliance, however, most configuration is via the pfSense web UI. The pfSense UI only listens on its LAN interface, so you cannot use it directly via the internet-facing floating IP address on its WAN interface.

1. Connect to the pfSense UI using SSH port forwarding to tunnel a connection through the jumpbox server connected to the internal network, onto the LAN interface of the pfSense appliance.

With a command line SSH client, you would typically run the following command:

```
ssh -i ~/.ssh/jumpbox.pem -l jumpboxuser -L 80:192.168.1.254:80 <jumpbox floating IP address>
```

where:

- 192.168.1.254 is the address allocated to the LAN interface of the appliance.

If you are using a Windows desktop, you may want to try using PuTTY for the SSH port forwarding.

2. Browse to <http://localhost/> to open the pfSense *Login* page.



Login to pfSense

Username

Password

pfSense is © 2004 - 2016 by Electric Sheep Fencing LLC. All Rights Reserved. [\[view license\]](#)

3. Log in with:

- **Username:** admin
- **Password:** If you're using the image provided by UKCloud, the default password is Password123#

Note! We recommend that you change the default password when you first login.

Set up your IPsec tunnel

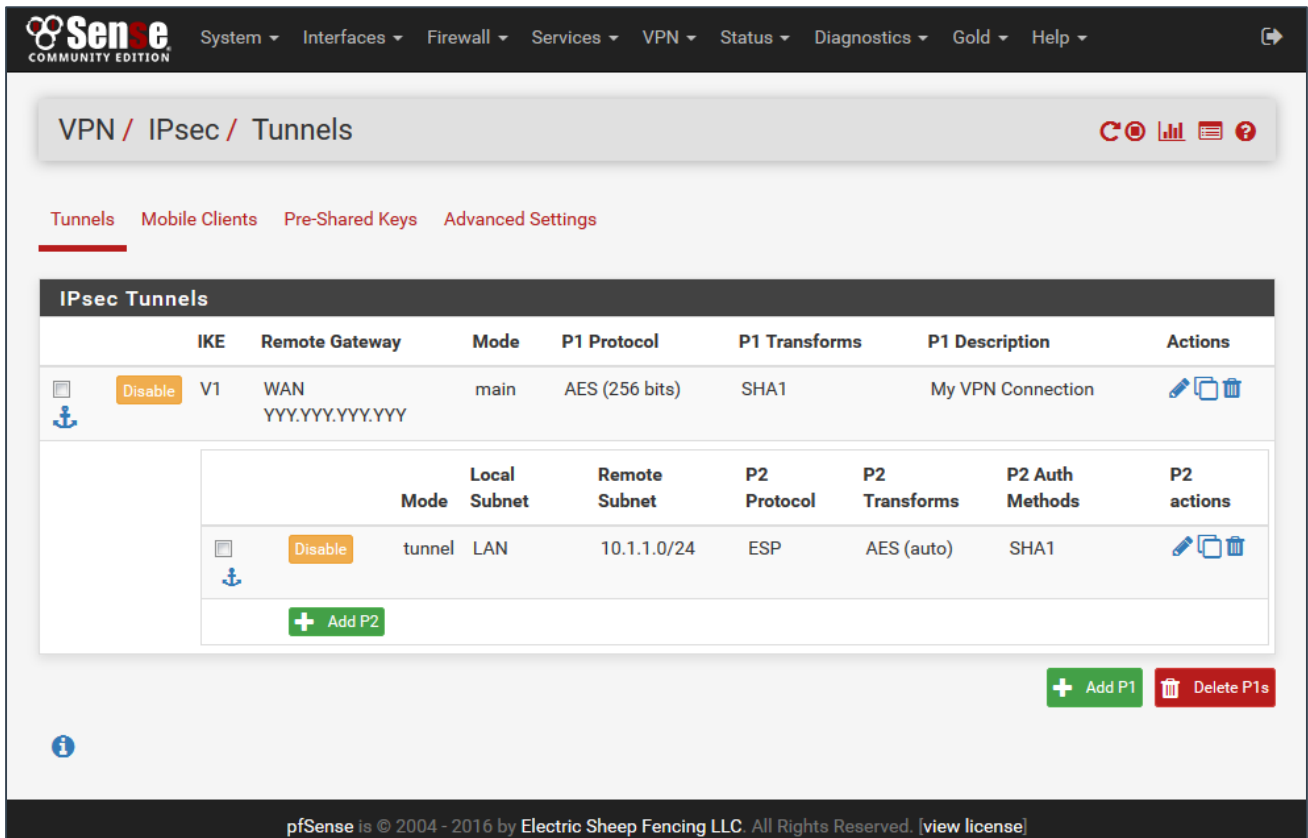
For the purpose of this blueprint, we're assuming the following tunnel configuration. You'll need to substitute your own values accordingly:

OpenStack Network	pfSense	VPN Endpoint	Remote Network
-----	-----	-----	-----
	LAN: 192.168.1.254	LAN: 10.1.1.1	
192.168.1.0/24 <---->	WAN: 172.16.1.XXX <----->	WAN: YYY.YYY.YYY.YYY <-->	10.1.1.0/24
	Floating IP: XXX.XXX.XXX.XXX		

Pre-shared Key: D3v0psD3m0D3v0psD3m0D3v0psD3m0D3v0psD3m0

1. From the menu, select **VPN > IPsec**.
2. On the *Tunnels* tab, click the **Add P1** button to start configuring this endpoint of the tunnel.
3. In the *Edit Phase 1* page, set the following values:
 - **Remote Gateway:** YYY.YYY.YYY.YYY
 - **Description:** My VPN Connection
 - **My identifier:** IP Address XXX.XXX.XXX.XXX
 - **Peer identifier:** IP Address YYY.YYY.YYY.YYY
 - **Pre-Shared Key:** D3v0psD3m0D3v0psD3m0D3v0psD3m0D3v0psD3m0
4. Save the settings.
5. Click the **Add P2** button to complete the rest of the tunnel configuration.
6. In the *Edit Phase 2* page, set the following values:
 - **Local Network:** LAN subnet
 - **NAT/BINAT translation:** None
 - **Remote Network:** Network 10.1.1.0/24
 - **Description:** My remote network
7. Save the settings.

The *Tunnels* page should now look something like the following:



Create a firewall rule to allow routing of returning traffic

The default firewall rules allow outbound traffic from the LAN network. This is sufficient to enable the IPsec tunnel to be established and to also allow traffic from the internal network to be routed across the tunnel. However, to allow returning traffic from the remote network to be routed back to the LAN network, you need to add a new firewall rule.

1. From the menu, select **Firewall > Rules** menu
2. Select the **IPsec** tab.

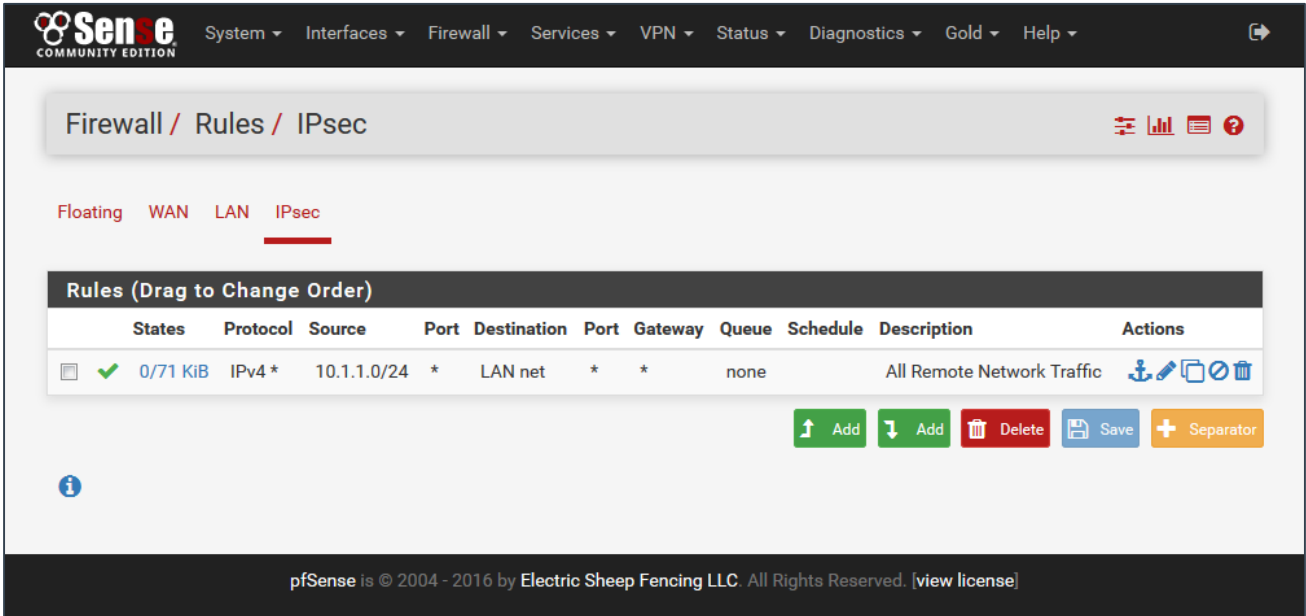
For this blueprint, since we are trusting all traffic from the LAN network to the remote network, we just need to add a single rule to accept all traffic from the remote network back into the LAN network. In your environment, you'll probably want to be more specific with your rule definitions.

3. Click one of the **Add** buttons and set the following values:
 - **Action:** Pass
 - **Interface:** IPsec
 - **Protocol:** any
 - **Source:** Network 10.1.1.0/24
 - **Destination:** LAN net

- **Description:** Allow all remote traffic

4. Save the rule.

The IPsec tab should now look something like the following:



Configure a static route for the remote network

The last thing to do to configure our OpenStack project to be able to connect to a remote network over the IPsec VPN is to update your router configuration to add a static route that redirects traffic for the remote network via the pfSense appliance. You can do this by editing the router in the Horizon UI, specifying the remote network and the LAN interface of the pfSense appliance:

Add Static Route ✕

Destination CIDR *

Next Hop *

Description:

Add static route to the router.
Next Hop IP must be a part of one of the subnets to which the router interfaces are connected.

Alternatively, you can define the same static route using the OpenStack CLI:

```
openstack router set --route destination=10.1.1.0/24,gateway=192.168.1.254 InternetGW
```

Configure the second endpoint

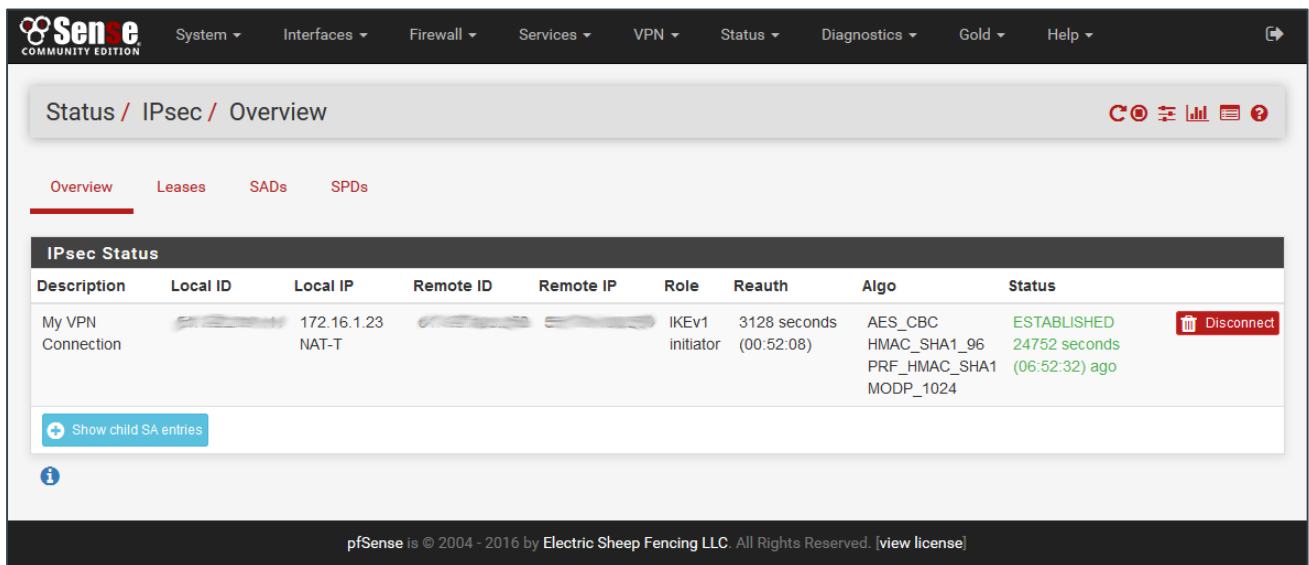
At this point, you now want to ensure that the other end of the VPN tunnel is also configured. If you are connecting two OpenStack projects, you can repeat the steps above to deploy another pfSense appliance into the remote project.

Connecting an IPsec tunnel to a vCNS Edge on a VDC in an Enterprise Compute Cloud environment can be problematic, as the vCloud Director web console does not set the local identifier correctly. You have to use a REST client against the vCloud Director API to manually correct the tunnel configuration. The process is documented in the [Changing IPsec VPN settings via that vCloud Director API](#) article on the UKCloud Portal Knowledge Centre.


However, we also provide a Ruby script in our GitHub repository that you can use, with the `vshield-configuration.yaml` sample configuration file, to automate the interaction with the vCloud Director API to correctly set up the VPN endpoint from scratch without having to use the web console. For more information about this script, see [here](#).

Check the tunnel status

By now, if the configuration of both endpoints match, the IPsec tunnel should already have completed both Phase 1 and Phase 2 negotiations and have established a tunnel. You can check by selecting **Status > IPsec** from the menu.



The screenshot shows the pfSense web interface. At the top, there is a navigation menu with items: System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, Gold, and Help. Below the menu, the breadcrumb path is 'Status / IPsec / Overview'. There are four tabs: Overview (selected), Leases, SADs, and SPDs. The main content area is titled 'IPsec Status' and contains a table with the following data:

Description	Local ID	Local IP	Remote ID	Remote IP	Role	Reauth	Algo	Status	
My VPN Connection	[blurred]	172.16.1.23 NAT-T	[blurred]	[blurred]	IKEv1 initiator	3128 seconds (00:52:08)	AES_CBC HMAC_SHA1_96 PRF_HMAC_SHA1 MODP_1024	ESTABLISHED 24752 seconds (06:52:32) ago	

Below the table, there is a button labeled 'Show child SA entries' and an information icon. At the bottom of the page, a footer reads: 'pfSense is © 2004 - 2016 by Electric Sheep Fencing LLC. All Rights Reserved. [view license]'.

About UKCloud

UKCloud has developed a range of cloud services designed specifically for the UK public sector, to help increase efficiencies, reduce costs, significantly improve procurement times and increase transparency. Our services are *easy to adopt, easy to use and easy to leave* to ensure that our customers remain in complete control with minimum risk. We were one of the first G-Cloud providers to achieve Pan Government Accreditation (PGA) up to Elevated OFFICIAL, and our services continue to achieve formal UK Government accreditations which make them suitable for all data at OFFICIAL (including OFFICIAL-SENSITIVE).

UKCloud's full offering consists of:

1. Infrastructure as a Service (IaaS) – seven offerings around Compute and Storage on demand
2. Software as a Service (SaaS) – offerings for email and collaboration
3. Platform as a Service (PaaS) – based upon Open Source Digital Application Platform and Hadoop which provides organisations the benefits of using a commodity cloud platform without the added management overheads

All of UKCloud's UK sovereign cloud computing services are hosted in one (or both) of our highly resilient tier 3 UK data centres in Farnborough and Corsham. UKCloud services are delivered with leading technologies from UKCloud Alliance Partners: QinetiQ, VMware, Cisco, EMC and Ark Data Centres. The Cloud Alliance also provides a collaborative resource which drives innovation and technical product development, helping to continually improve UKCloud's offering to meet the needs of the UK public sector.

UKCloud is focused on providing cloud services in a more agile, secure and cost effective manner. We strive to deliver solutions that harness technology as a way to facilitate the changes that are needed to streamline processes and reduce costs to support the UK public sector and, ultimately, UK citizens and taxpayers.

MORE INFORMATION ►

For further information about UKCloud and how we can help you, send an email to info@ukcloud.com

UKCloud Ltd

A8 Cody Technology Park
Ively Road, Farnborough
Hampshire, GU14 0LX

T 01252 303300

E: info@ukcloud.com

ukcloud.com



[@ukcloudltd](https://twitter.com/ukcloudltd)



[ukcloudltd](https://www.facebook.com/ukcloudltd)



[ukcloud-ltd](https://www.linkedin.com/company/ukcloud-ltd)

Reasonable efforts have been made to ensure the accuracy of the information contained in this document. No advice given or statements or recommendations made shall in any circumstances constitute or be deemed to constitute a warranty by UKCloud Ltd as to the accuracy of such advice, statements or recommendations. UKCloud Ltd shall not be liable for any loss, expense, damage or claim howsoever arising out of the advice given or not given or statements made or omitted to be made in connection with this document.

No part of this document may be copied, reproduced, adapted or redistributed in any form or by any means without the express prior written consent of UKCloud Ltd.

**© UKCloud Ltd 2017
All Rights Reserved.**

UKC-GEN-476 • 01/2017