|  |  |
| --- | --- |
| **cloudenabled** | |
| **Author** | **Sushil Kumar** |
| **Documentation Version** | **V1.1** |

**Pivotal Cloud Foundry v1.10**

**Installing Pivotal Cloud Foundry on AWS**

Ref: <https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform.html/>

This guide describes how to install [Pivotal Cloud Foundry](https://network.pivotal.io/products/pivotal-cf) (PCF) on Amazon Web Services (AWS) using the PCF CloudFormation template.

**Topics:**

1. **Prerequisites**
   1. **General Requirements**
   2. **AWS Requirements**
2. **Install PCF using CloudFormation**

**Why CloudFormation:**

The CloudFormation template for PCF describes the set of necessary AWS resources and properties. When you create an AWS stack using the PCF template, CloudFormation provisions all the infrastructure that you need to deploy PCF on AWS.

1. **Prerequisites**

The following sections describe general requirements for running PCF and specific requirements for running PCF on AWS.

**General Requirements**

The following are general requirements for deploying and managing a PCF deployment with

Ops Manager and

Elastic Runtime:

* Ability to create a wildcard DNS record to point to your router or load balancer.
* Elastic Runtime gives each application its own hostname in your app domain
* Sufficient IP allocation:
  + - IPs needed = VM instances + static IPs + errands + compilation workers
* The most recent version of the Cloud Foundry Command Line Interface (cf CLI)
* One or more NTP servers if not already provided by your IaaS

**AWS Requirements**

* 1 Elastic Load Balancer
* 1 Relational Database Service. We recommend at least a db.m3.xlarge instance with 100 GB of allocated storage.
* 5 S3 Buckets
* EC2 Instances:
  + - 13 t2.micro
    - 15 t2.small
    - 2 m3.medium
    - 6 m3.xlarge
    - 3 m3.2xlarge

1. **Install PCF on AWS using CloudFormation**
   1. [Deploying the CloudFormation Template for PCF on AWS](https://docs.pivotal.io/pivotalcf/1-7/customizing/cloudform-template.html)
   2. [Launching an Ops Manager Director Instance on AWS](https://docs.pivotal.io/pivotalcf/1-7/customizing/cloudform-om-deploy.html)
   3. [Configuring Ops Manager Director on AWS](https://docs.pivotal.io/pivotalcf/1-7/customizing/cloudform-om-config.html)
   4. [Deploying Elastic Runtime on AWS](https://docs.pivotal.io/pivotalcf/1-7/customizing/cloudform-er-config.html)
2. **Deploying the CloudFormation Template for Pivotal Cloud Foundry on AWS**
3. Step 1: Download the PCF CloudFormation Template
4. Step 2: Upload an SSL Certificate to AWS
   1. (Option) Create SSL Certificate using the AWS CLI
   2. (Option) Create SSL Certificate using the AWS Certificate Manager
5. Step 3: Create a Resource Stack Using the CloudFormation Template

The template is designed to output the resources necessary for two availability zones (AZ), with a private and public subnet designated for each AZ.

The Elastic Load Balancer will be attached to the public subnet of both AZs to balance traffic across both environments

**Step 1: Download the PCF CloudFormation Template:**

1. Sign in to [Pivotal Network](https://network.pivotal.io/).
2. Select **Elastic Runtime**. From the **Releases** drop-down menu, select the release that you wish to install.
3. Download the **PCF 1.7 CloudFormation script for AWS**.
4. Save the file as pcf.json.

**Step 2: Upload an SSL Certificate to AWS**

* The [AWS Certificate Manager](https://docs.pivotal.io/pivotalcf/1-7/customizing/cloudform-template.html#aws-certificate-manager)
* Create a certificate, must use your system wildcard domain (example: \*.my-pcf-apps-domain.com) as the Common Name input.
  + Ref: \*.e-cloudenabled.com
* Add the following additional domains and wildcards using OpenSSL’s SAN (subjectAltName) extension: \*.system.yourdomain.com, \*.login.system.yourdomain.com, \*.uaa.system.yourdomain.com, and \*.apps.yourdomain.com

Ref:

\*.system.e-cloudenabled.com,

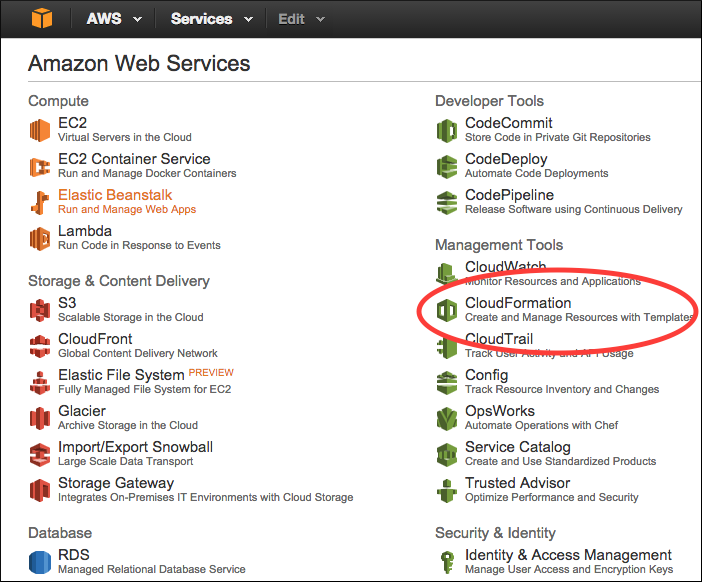
\*.login.system.e-cloudenabled.com,

\*.uaa.system.e-cloudenabled.com,

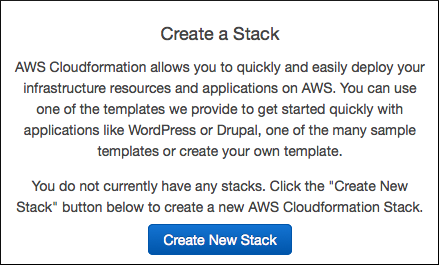
\*.apps.e-cloudenabled.com

**Step 3: Create a Resource Stack Using the CloudFormation Template**

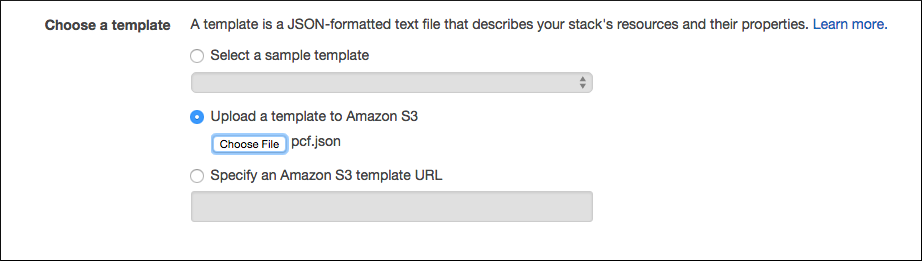
1. Log in to the [AWS Console](https://console.aws.amazon.com/).
2. In the second column, under **Management Tools**, click **CloudFormation**.



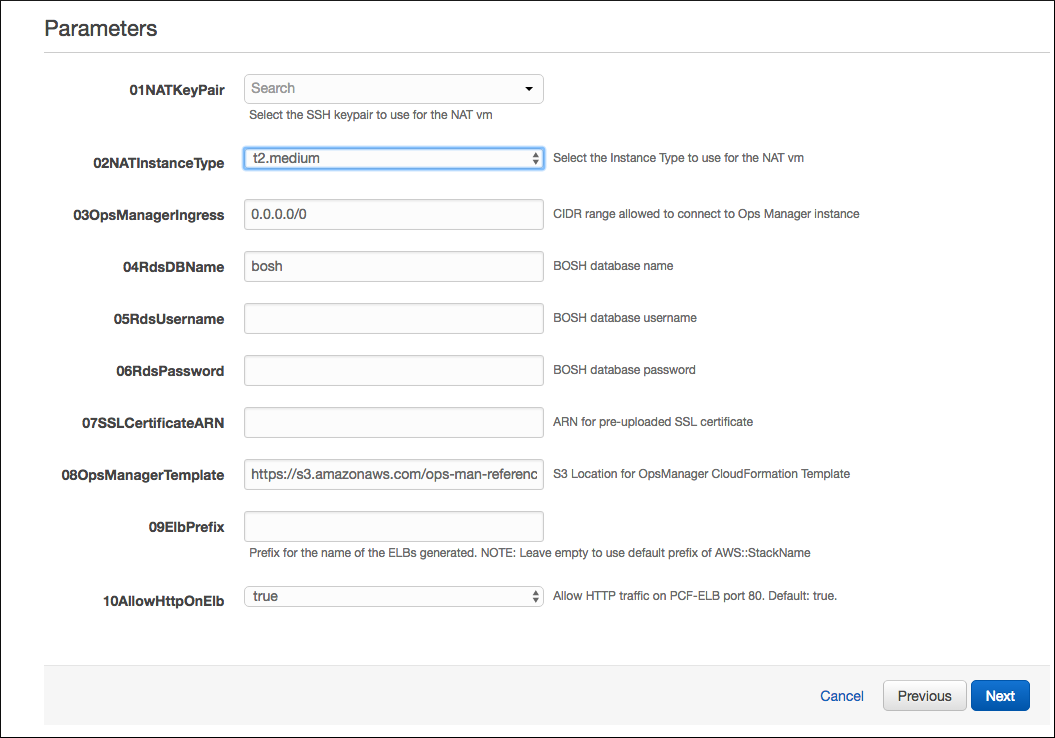
1. Click **Create New Stack**.



1. Select **Upload a template to Amazon S3**.



1. Click **Browse**. Browse to and select the pcf.json, the **Pivotal Cloud Foundry CloudFormation script for AWS** file that you downloaded. Click **Next**.
2. On the next screen, name the stack pcf-stack.
3. In the **Specify Parameters** page, complete the following fields:

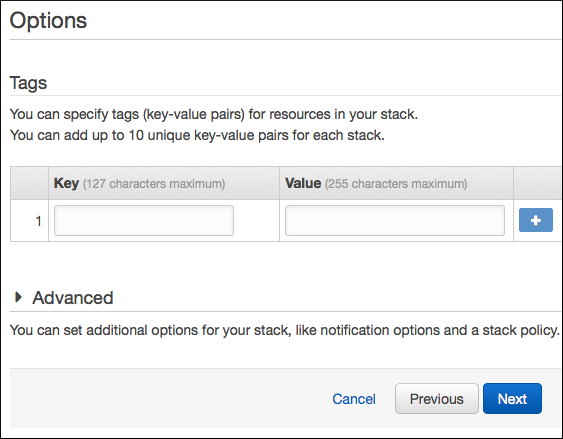


* + **01NATKeyPair**: Use the drop-down menu to select the name of your pre-existing AWS key pair.
  + **02NATInstanceType**: Do not change this value.
  + **03OpsManagerIngress**: Do not change this value.
  + **04RdsDBName**: Do not change this value.
  + **05RdsUserName**: Enter a username for the RDS database. : sushil

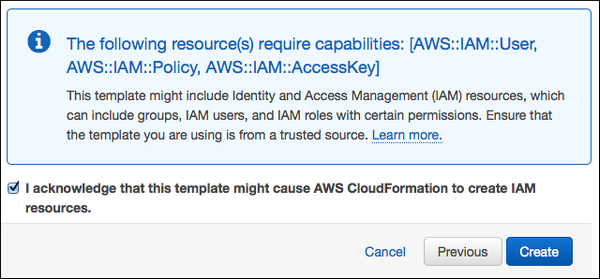
**Note**: Do not enter the username rdsadmin. AWS reserves the rdsadmin user account for internal database instance management.

* + **06RdsPassword**: Enter a password for the RDS database. : sushil123
  + **07SSLCertificateARN**: Enter your uploaded SSL Certificate ARN.
  + **08OpsManagerTemplate**: The default template link provided here works.
  + **09ElbPrefix**: Prefix for the generated names of the ELBs. Any string you specify in this field will be prefixed to -pcf-elb to form the name of your ELBs. Leave empty to use the default prefix of AWS::StackName.
  + **10AllowHttpOnElb**: Set this to true to listen for HTTP traffic on port 80. This is the default. Set it to false to only listen for traffic on ports 443 and 4443.

1. Click **Next**.
2. On the **Options** page, leave the fields blank and click **Next**.

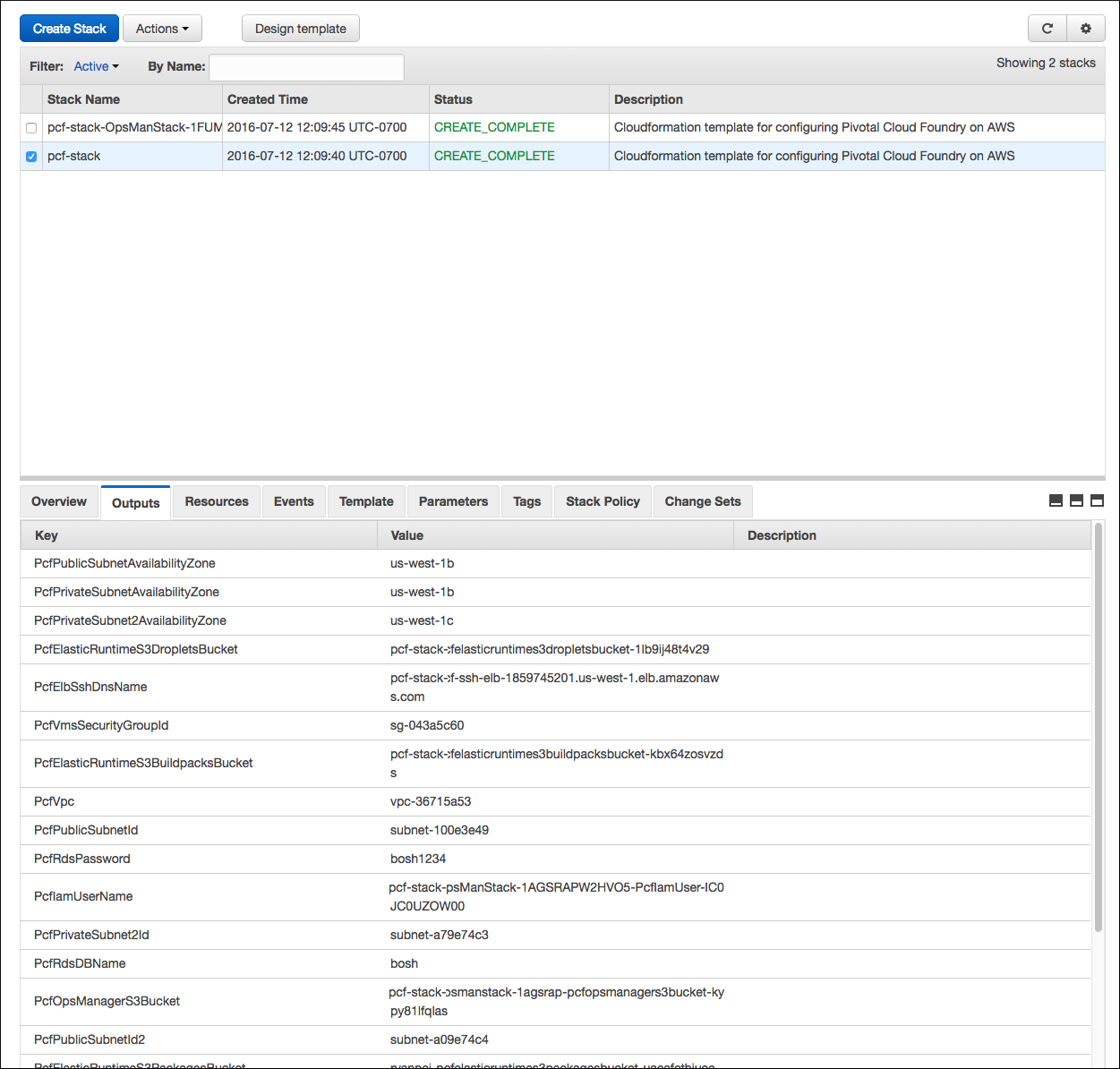


1. On the Review page, select the I acknowledge that this template might cause AWS CloudFormation to create IAM resources checkbox and click Create.



AWS runs the CloudFormation script and creates the infrastructure that you need to deploy PCF to AWS. This may take a few moments. You can click on the **Events** tab to view the progress of the setup.

When the installation process successfully completes, AWS displays **CREATE\_COMPLETE** as the status of the stack.



1. **Launching an Ops Manager Director Instance on AWS**

[Step 1: Open the Outputs Tab in AWS Stacks](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#outputs)

[Step 2: Select a Pivotal Ops Manager AMI Instance](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#select-ami)

[Step 3: Configure Instance Details](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#configure-instance)

[Step 4: Configure Security Group](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#security-group)

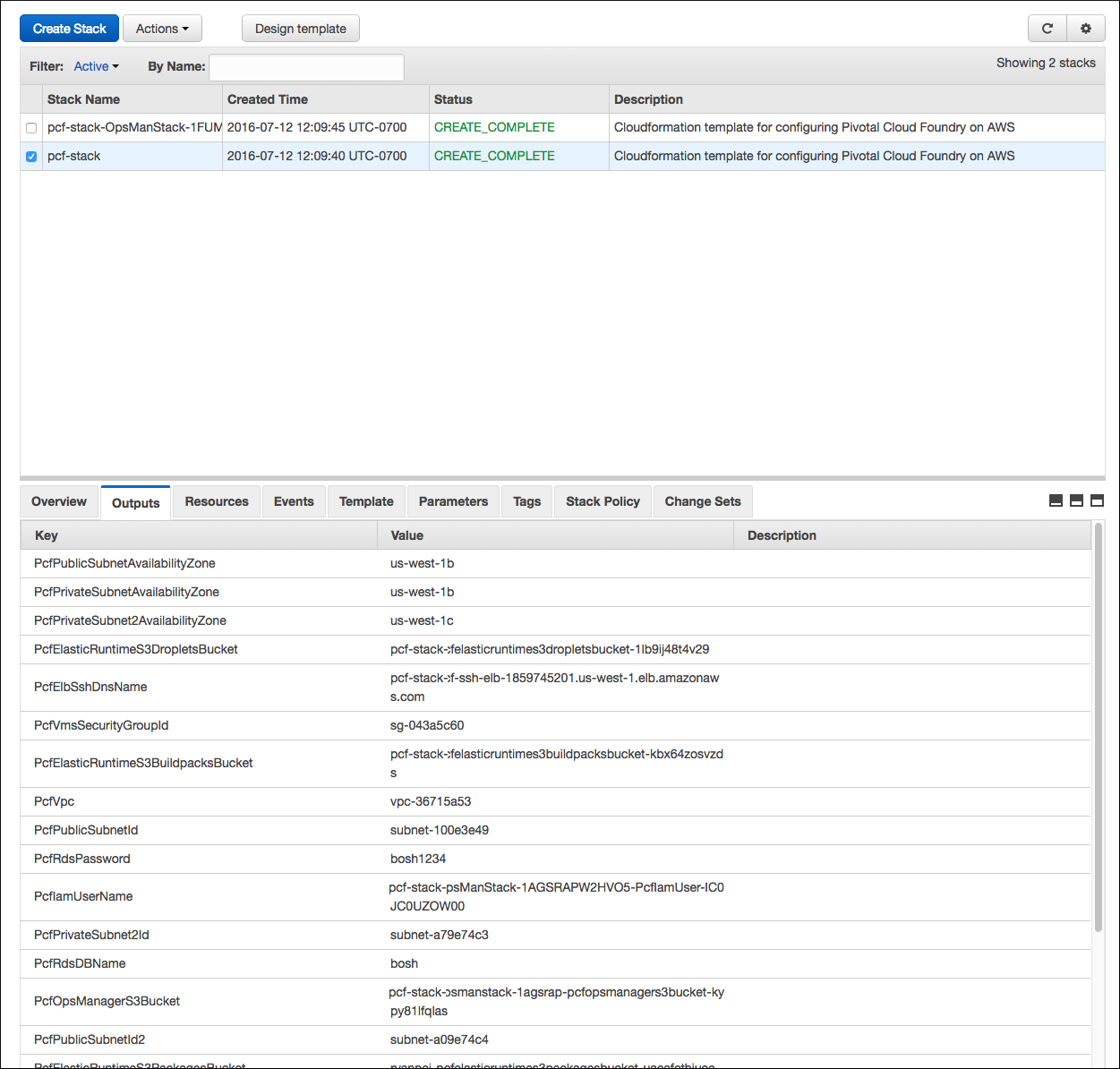
[Step 5: Deploy Ops Manager](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#deploy)

[Step 6: Create a DNS Entry](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#create-dns)

[Step 7: Configure Ops Manager Director for AWS](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#configure)

[Step 1: Open the Outputs Tab in AWS Stacks](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#outputs):

1. In the dashboard of your [AWS Console](https://console.aws.amazon.com/), click **CloudFormation**. The Stacks Dashboard displays.
2. Select the **pcf-stack** checkbox, then select the **Outputs** tab.



In the steps described below, use the information from the **Value** column of the **Outputs** tab to configure your PCF installation.

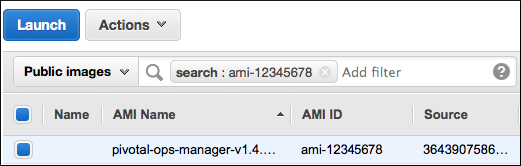
Step 2: Select a Pivotal Ops Manager AMI Instance

1. Log in to the [Pivotal Network](https://network.pivotal.io/products/pivotal-cf) and click Pivotal Cloud Foundry **Ops Manager**.
2. From the **Releases** dropdown, select the release to install.
3. Select **Pivotal Cloud Foundry Ops Manager for AWS** to download the OpsManagerx.x.xonAWSFulfillmentInstructions.pdf file. This document lists AMI IDs for Pivotal Ops Manager for specific regions.

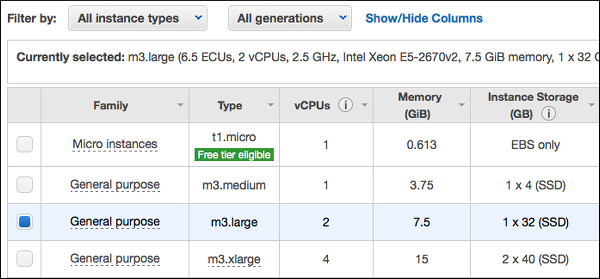


Ireland region AMI used: ami-5baaaa3d

1. Log in to the [AWS Console](https://console.aws.amazon.com/). Navigate to the EC2 Dashboard.
2. In the left navigation panel, click **AMIs**.
3. Using the OpsManagerx.x.xonAWSFulfillmentInstructions.pdf document, enter the AMI ID for your AWS region in the Public images search field.



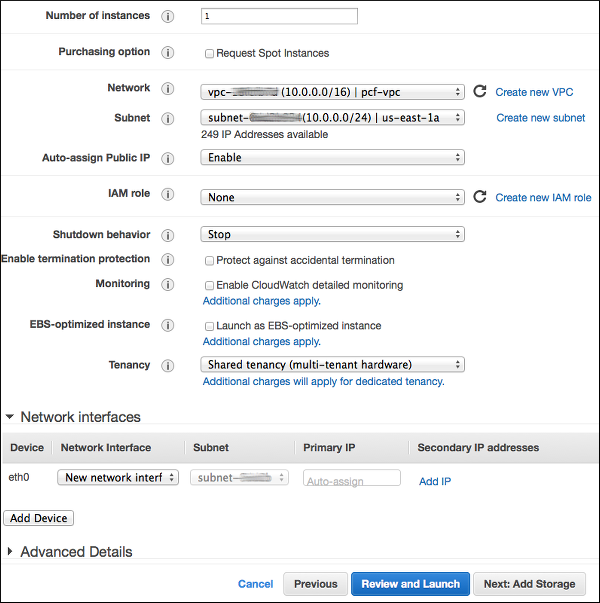
1. Select this AMI and click **Launch**.
2. Choose **m3.large** for your instance type.



1. Click **Next: Configure Instance Details**.

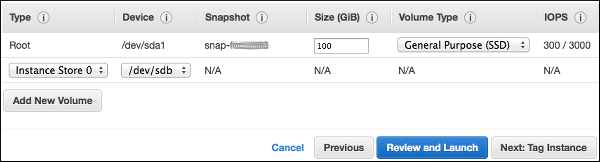
Step 3: Configure Instance Details

1. Complete the **Config Instance Details** page with information from the [Outputs tab](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#outputs) in the AWS Stacks Dashboard:

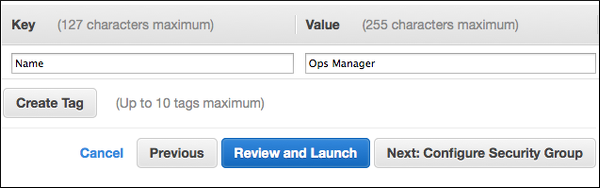


* + Select the **Network** that matches the value of **PcfVpc**.
  + Select the **Subnet** that matches the value of **PcfPublicSubnetId**.

1. Set **Auto-assign Public IP** to **Enable**.
2. Click **Next: Add Storage**.
3. On the **Add Storage** page, adjust the **Size (GiB)** value. Pivotal recommends increasing this value to a minimum of 100 GB.

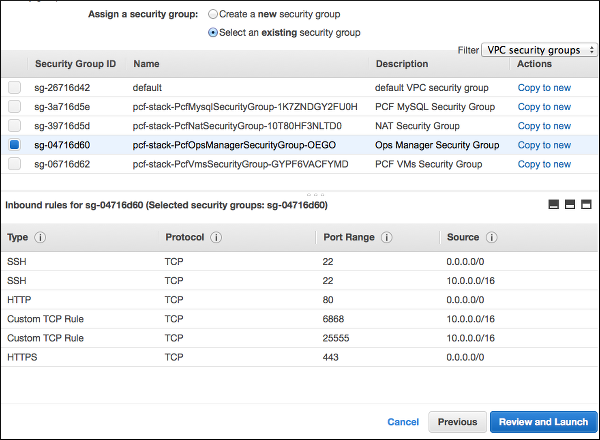


1. Click **Next: Tag Instance**.
2. On the **Tag Instance** page, add a **Key** Name with **Value** Ops Manager.



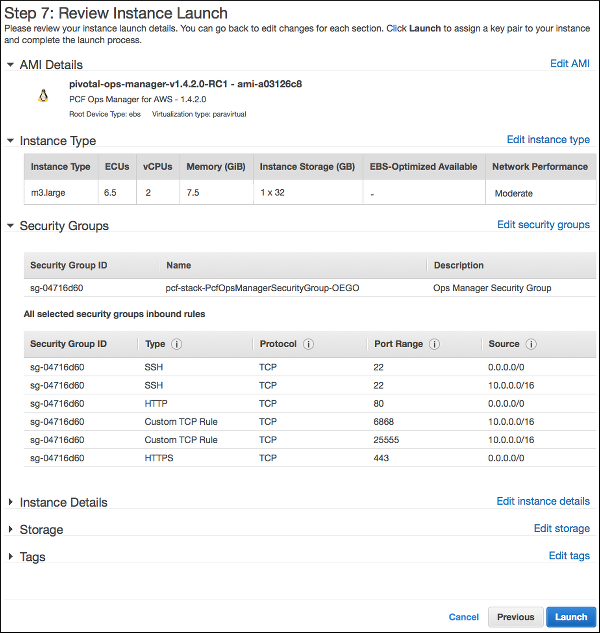
1. Click **Next: Configure Security Group**.

Step 4: Configure Security Group

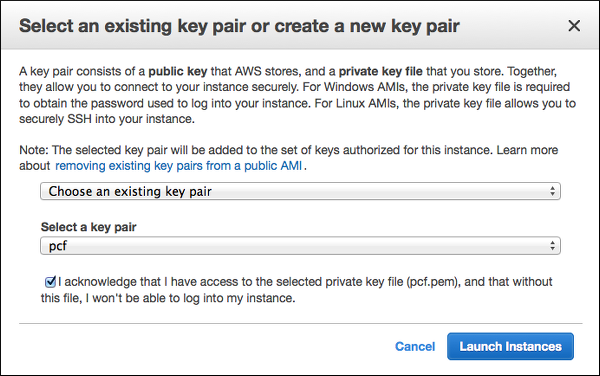
1. Select the **Select an existing security group** option.
2. Select the **Security Group ID** that matches the value of **PcfOpsManagerSecurityGroupId** located in the [Outputs](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#outputs) tab of the Stacks dashboard. 
3. Click **Review and Launch**.

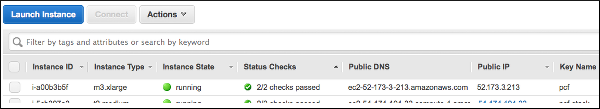
Step 5: Deploy Ops Manager

1. Review the instance launch details. Click **Launch**.



1. Use the first drop-down menu to select **Choose an existing key pair**. Use the second drop-down menu to select the name of your pre-existing AWS key pair.
2. Select the acknowledgement checkbox.
3. Click **Launch Instances**. If successful, you will see the **Launch Status Page**.



1. Click **View Instances**. Or alternately, navigate to **Instances** from the left navigation panel of the EC2 Dashboard.
2. AWS deploys Ops Manager. This may take a few minutes. When complete, AWS displays an **Instance State** of running and a **Status Check** of passed when the Ops Manager deployment successfully completes. 

Step 6: Create a DNS Entry

Create a DNS entry for the IP address that you used for Ops Manager. You must use this fully qualified domain name when you log into Ops Manager in the Configure Ops Manager Director for AWS step below.

1. **Step 7: Configure Ops Manager Director for AWS**

Step 1: Open the Outputs Tab in AWS Stacks

Step 2: Access Ops Manager

Use an Identity Provider (IdP)

Internal Authentication

Step 3: AWS Config Page

Step 4: Director Config Page

Step 5: Create Availability Zones Page

Step 6: Create Networks Page

Step 7: Assign AZs and Networks Page

Step 8: Security Page

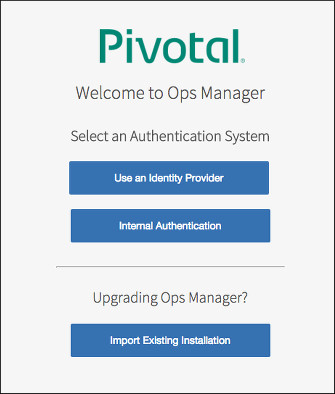
Step 9: Resource Config Page

Step 10: Complete the Ops Manager Director Installation

## Step 1: Open the Outputs Tab in AWS Stacks

## Step 2: Access Ops Manager

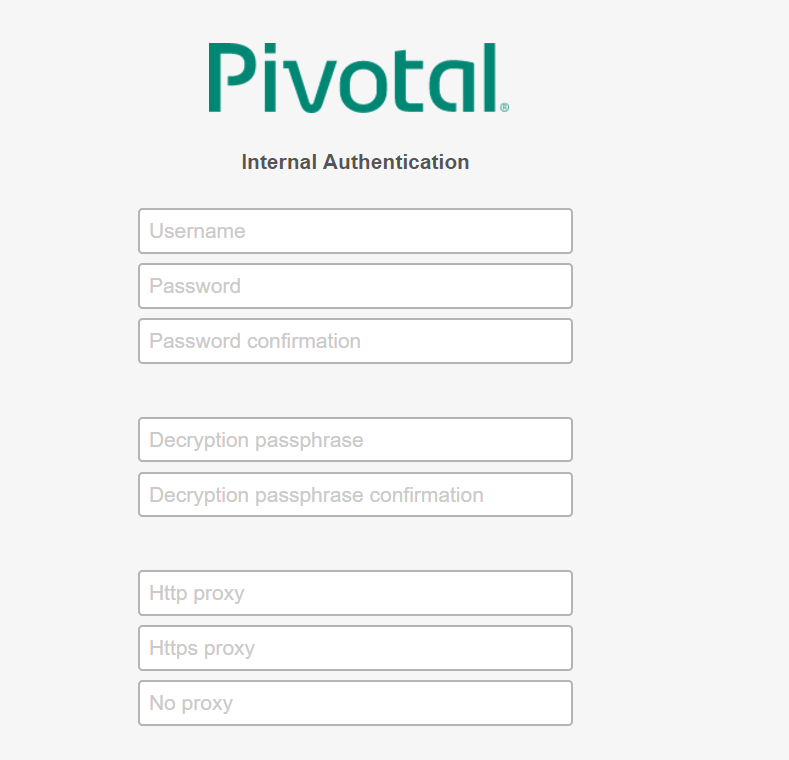
In a web browser, navigate to the fully qualified domain of Ops Manager Director Instance on AWS.



Use Internal Authentication:

When redirected to the **Internal Authentication** page, you must complete the following steps:

* Enter a **Username**, **Password**, and **Password confirmation** to create an Admin user.
* Enter a **Decryption passphrase** and the **Decryption passphrase confirmation**. This passphrase encrypts the Ops Manager datastore, and is not recoverable if lost.
* For  **HTTP proxy** or **HTTPS proxy**, leave blank
* Read the **End User License Agreement**, and select the checkbox to accept the terms.
* Click **Setup Authentication**



Uname : sushil

Pwd: sushil123

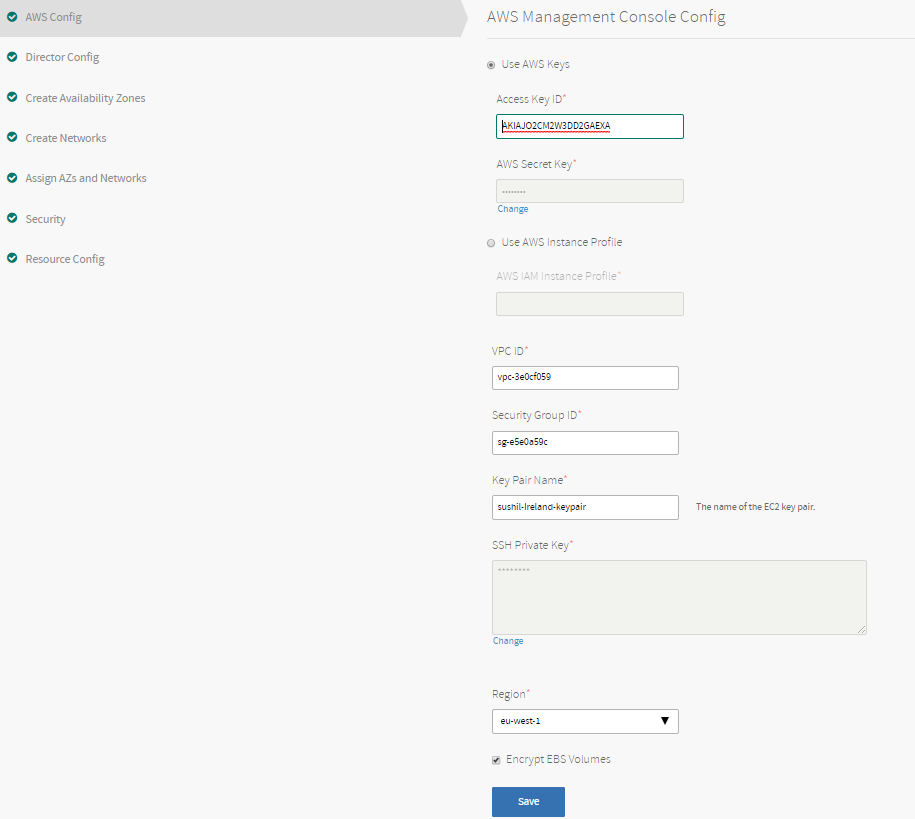
Passphrase: sushil

## Step 3: AWS Config Page

1. Click the **Ops Manager Director** tile.



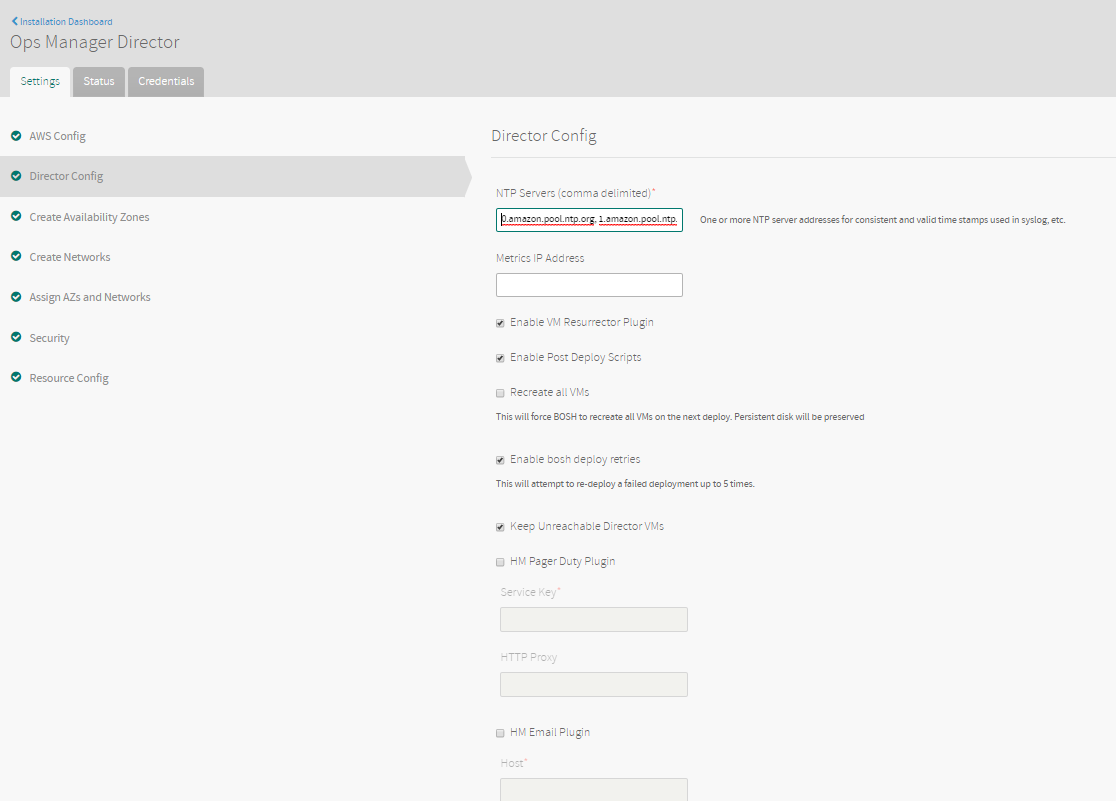
1. Select **AWS Config** to open the **AWS Management Console Config** page.

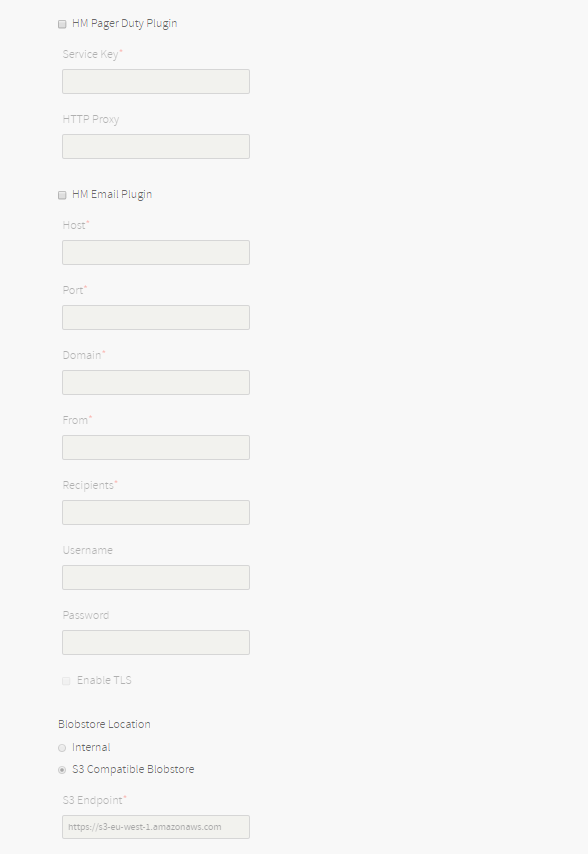


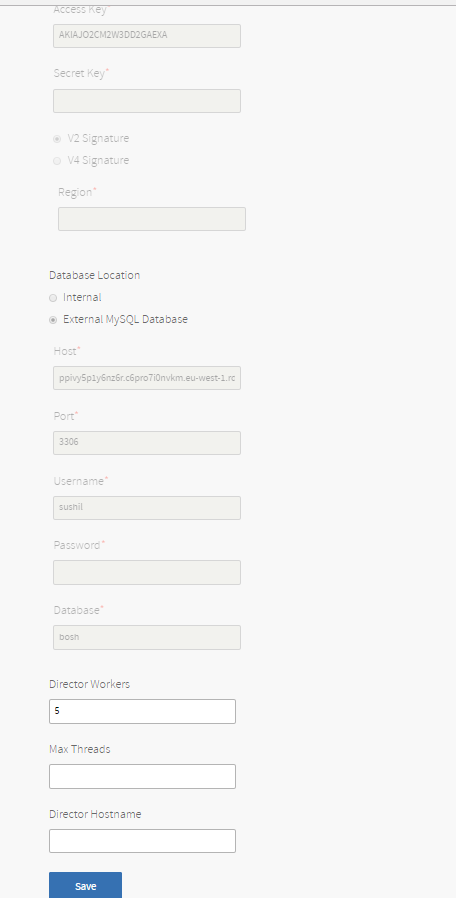
1. Select **Use AWS Keys** or **Use AWS Instance Profile**.
   * If you choose to use AWS keys, complete the fields with information from the [**Outputs**](http://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-config.html#outputs) tab for your stack in the AWS Console:
     + - **Access Key ID**: Use the value of **PcfIamUserAccessKey**.
       - **AWS Secret Key**: Use the value of **PcfIamUserSecretAccessKey**.
2. Complete the remainder of the **AWS Management Console Config** page with the following information.
   * + **VPC ID**: Use the value of **PcfVpc** from your **Outputs** tab.
     + **Security Group ID**: Open the AWS EC2 Dashboard and click **Security Groups**. Select the security group with the **Description** PCF VMs Security Group. Copy the **Group ID** of this group into the Ops Manager **Security Group ID** field.
     + **Key Pair Name**: Use the name of your pre-existing AWS key pair. You selected this key pair name when you first [deployed Ops Manager Director](http://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#deploy).
     + **SSH Private Key**: Open your AWS key pair .pem file in a text editor. Copy the contents of the .pem file and paste it into the **SSH Private Key** field.
     + **Region**: Select the region where you deployed Ops Manager.
     + **Encrypt EBS Volumes**: Select this checkbox to enable full encryption on persistent disks of all BOSH-deployed virtual machines (VMs), except for the Ops Manager VM and Director VM. See the [Configuring Amazon EBS Encryption for PCF on AWS](http://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-ebs-config.html) topic for details on using EBS encryption.
3. Click **Save**.

## Step 4: Director Config Page

1. Select **Director Config** to open the **Director Config** page.



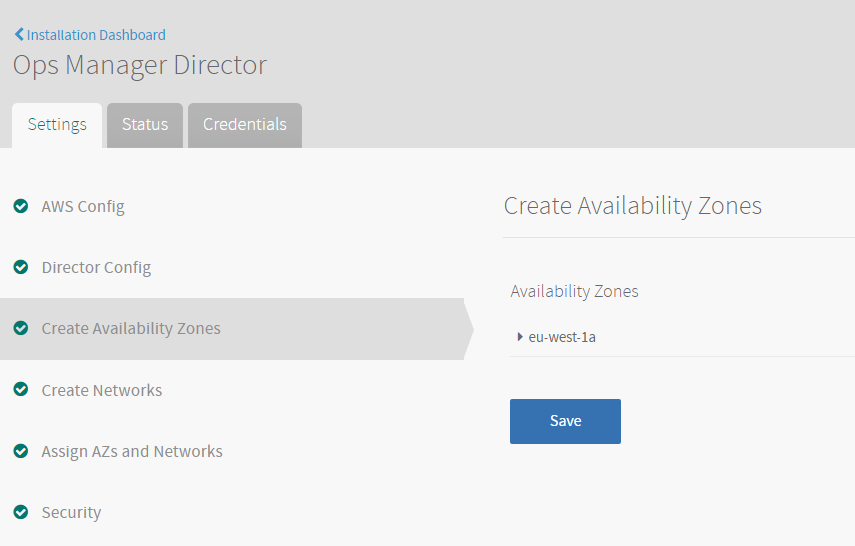




1. In the **NTP Servers (comma delimited)** field, enter at least two of the following NTP servers, separated by a comma:
   * 0.amazon.pool.ntp.org
   * 1.amazon.pool.ntp.org
   * 2.amazon.pool.ntp.org
   * 3.amazon.pool.ntp.org
2. Enter your **Metrics IP Address** blank
3. Select the **Enable VM Resurrector Plugin** checkbox to enable the Ops Manager Resurrector functionality and increase Elastic Runtime availability.
4. Select **Enable Post Deploy Scripts** to run a post-deploy script after deployment. This script allows the job to execute additional commands against a deployment.
5. Select **Recreate all VMs** to force BOSH to recreate all VMs on the next deploy. This process does not destroy any persistent disk data.
6. Select **Enable bosh deploy retries** if you want Ops Manager to retry failed BOSH operations up to five times.
7. Select **Keep Unreachable Director VMs** if you want to preserve Ops Manager Director VMs after a failed deployment for troubleshooting purposes.
8. DeSelect **HM Pager Duty Plugin** to enable Health Monitor integration with PagerDuty.
9. Select **HM Email Plugin** to enable Health Monitor integration with email.
10. For **Blobstore Location**, select **S3 Compatible Blobstore** and complete the following steps:
    * In a browser, reference the [**Amazon Simple Storage Service (Amazon S3) table**](http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region), and find the region for your AWS account.
    * Prepend https:// to the **Endpoint** for your region, and copy it into the Ops Manager **S3 Endpoint** field. For example, in the **us-west-2** region, enter https://s3-us-west-2.amazonaws.com into the field.
    * Complete the following fields with information from the **Outputs** tab in the AWS Console:
      + **Bucket Name**: Use the value of **PcfOpsManagerS3Bucket**.
      + **Access Key ID**: Use the value of **PcfIamUserAccessKey**.
      + **AWS Secret Key**: Use the value of **PcfIamUserSecretAccessKey**.
    * Select **V2 Signature**
11. For **Database Location**, select **External MySQL Database**. Complete the following fields with information from the **Outputs** tab in the AWS Console.
    * + **Host**: Use the value of **PcfRdsAddress**.
      + **Port**: Use the value of **PcfRdsPort**.
      + **Username**: Use the value of **PcfRdsUsername**.
      + **Password**: Use the value of **PcfRdsPassword**.
      + **Database**: Use the value of **PcfRdsDBName**.
12. **Max Threads** leave blank
13. **Director Hostname leave blank**.
14. Click **Save**.

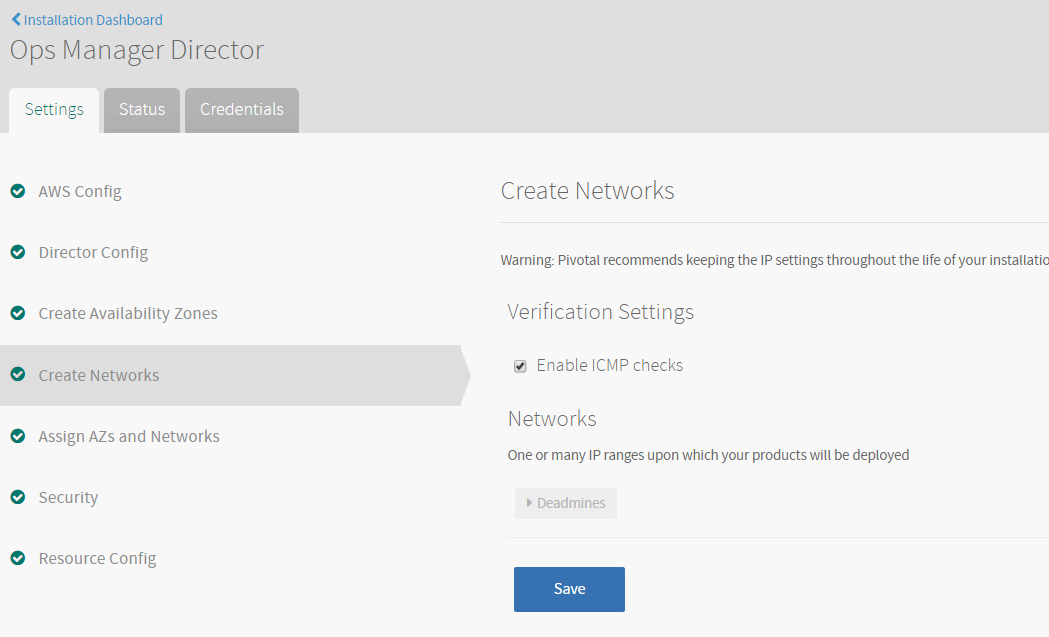
## Step 5: Create Availability Zones Page

1. Select **Create Availability Zones**.

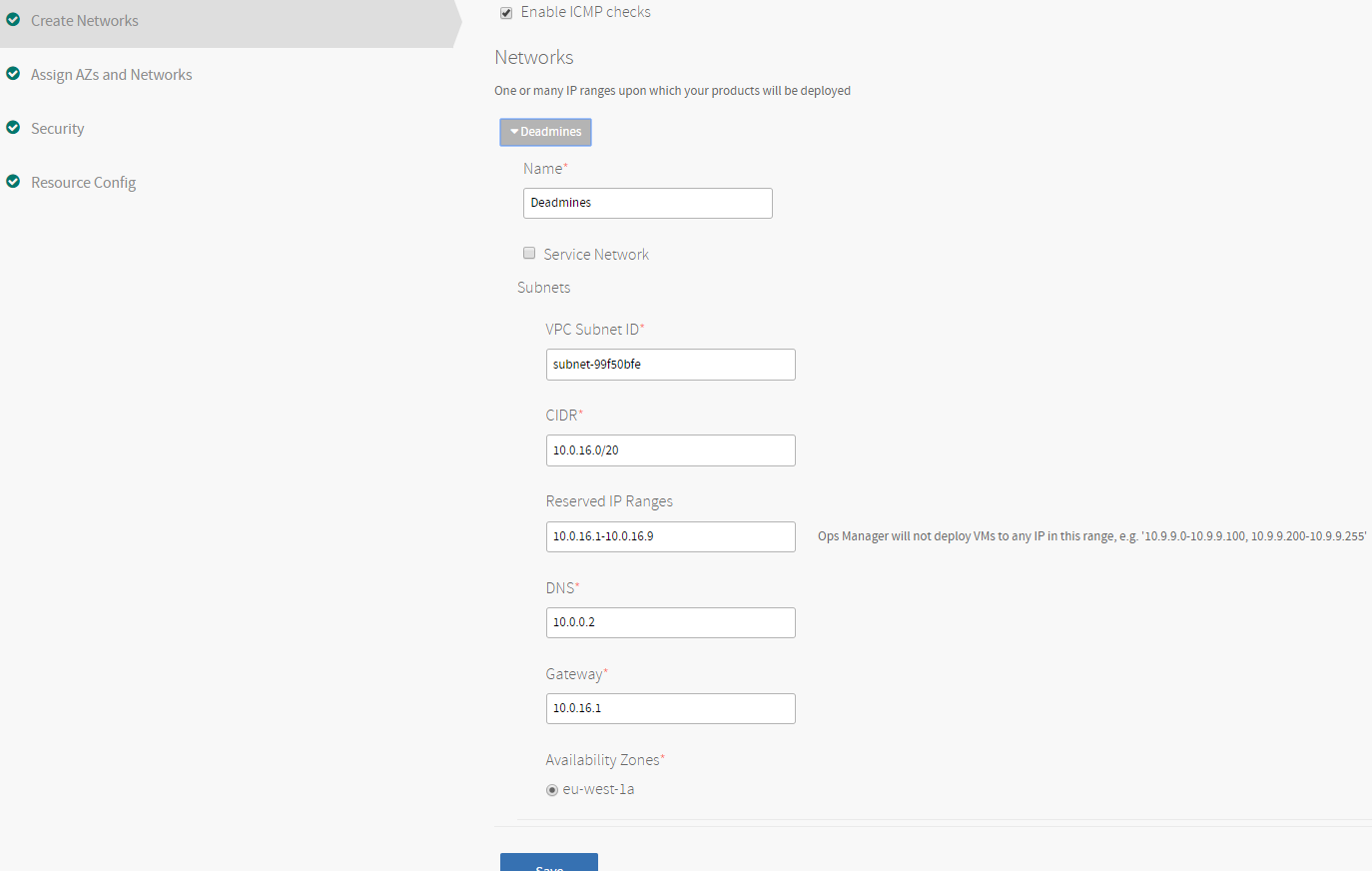


1. Use the following steps to create one or more Availability Zones for your applications to use:
   * Click **Add**.
   * For **Amazon Availability Zone**, enter the value of **PcfPrivateSubnetAvailabilityZone** from the **Outputs** tab in the AWS Console.
   * **(Optional)** If you are using a second **Amazon Availability Zone**, click **Add**. Enter the value of **PcfPrivateSubnet2AvailabilityZone** from the **Outputs** tab in the AWS Console.
   * Click **Save**.

Step 6: Create Networks Page

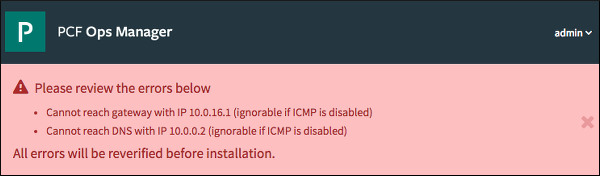


1. Select **Create Networks**.
2. Select **Enable ICMP checks** to enable ICMP on your networks. Ops Manager uses ICMP checks to confirm that components within your network are reachable.
3. Use the following steps to create one or more Ops Manager networks:



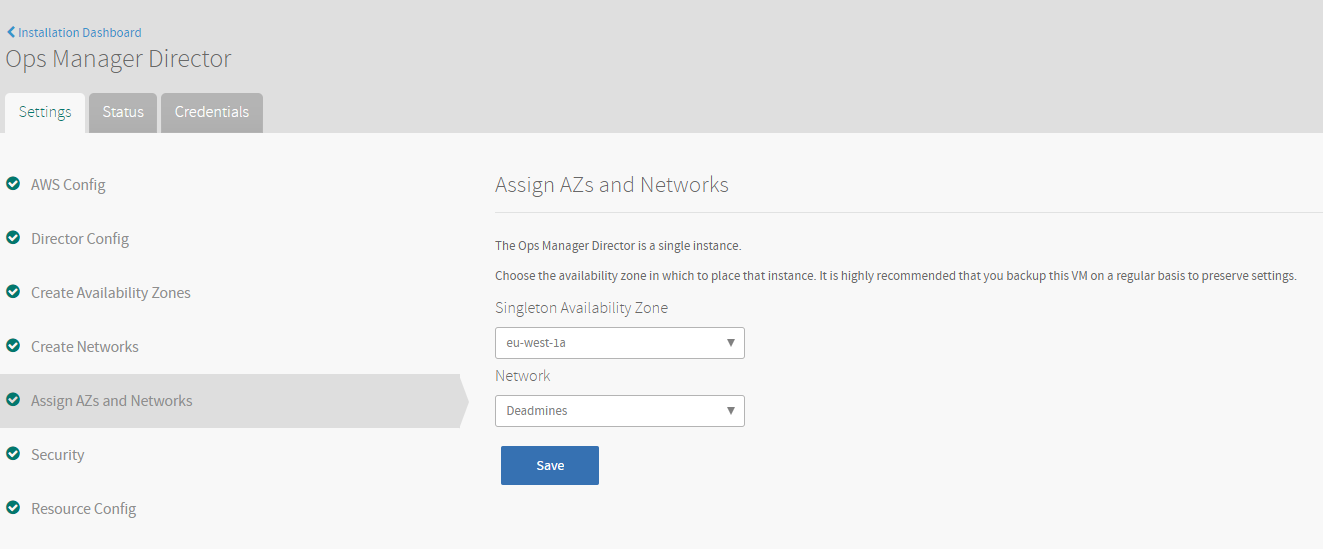
* + Click **Add Network**.
  + Enter a unique **Name** for the network.
  + If you want to dynamically provision VMs in this network for use with on-demand services, select the **Service Networks** checkbox. When the checkbox is selected, Ops Manager does not provision VMs within the specified CIDR range.
  + Click **Add Subnet** to create one or more subnets for the network.
  + In the **VPC Subnet ID** field, use the value of **PcfPrivateSubnetId** from the **Outputs** tab in the AWS Console.
  + For **CIDR**, enter 10.0.16.0/20. Ops Manager deploy VMs to this CIDR block.
  + For **Reserved IP Ranges**, enter 10.0.16.1-10.0.16.9. Ops Manager avoids deploying VMs to any IP address in this range.
  + Enter 10.0.0.2 for **DNS** and 10.0.16.1 for **Gateway**.
  + Select which **Availability Zones** to use with the network.

1. Click **Save**.
2. If the following ICMP error message appears, you can ignore the warning. Dismiss the warning, and move on to the next step.



Step 7: Assign AZs and Networks Page

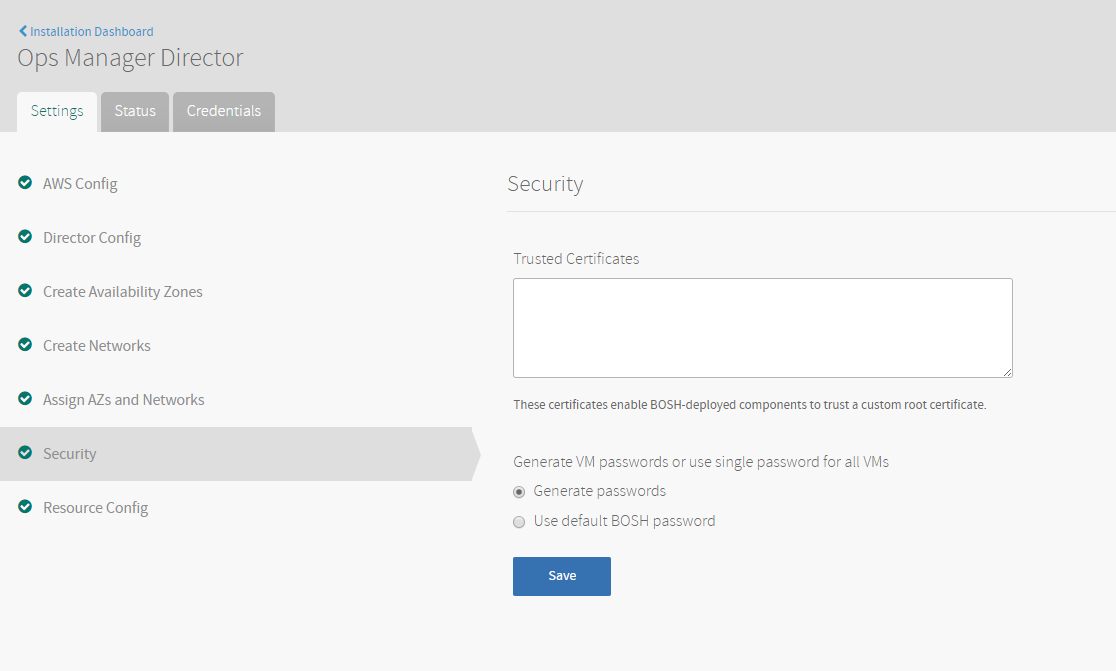
1. Select **Assign AZs and Networks**.



1. Use the drop-down menu to select a **Singleton Availability Zone**. The Ops Manager Director installs in this Availability Zone.
2. Use the drop-down menu to select a **Network** for your Ops Manager Director.
3. Click **Save**.

Step 8: Security Page

1. Select **Security**.



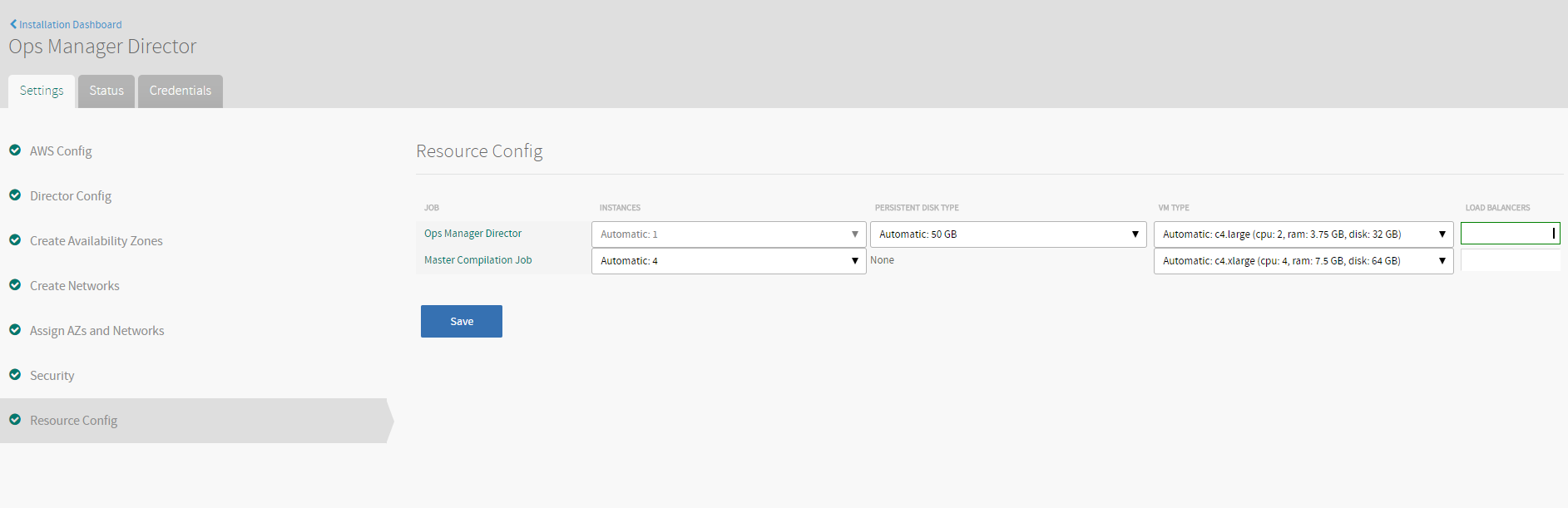
1. In **Trusted Certificates**, enter a custom certificate authority (CA) certificate to insert into your organization’s certificate trust chain. This feature enables all BOSH-deployed components in your deployment to trust a custom root certificate. If you want to use Docker Registries for running app instances in Docker containers, use this field to enter your certificate for your private Docker Registry. See the [Using Docker Registries](http://docs.pivotal.io/pivotalcf/1-10/opsguide/docker-registry.html) topic for more information.

**Here**: we are leaving blank

1. Choose **Generate passwords** .
2. Click **Save**. To view your saved Director password, click the **Credentials** tab.

Step 9: Resource Config Page

1. Select **Resource config**.

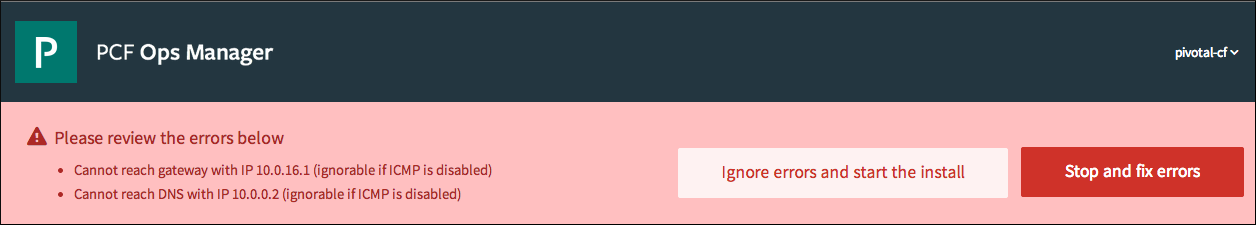


1. Adjust any values as necessary for your deployment. Under the **Instances**, **Persistent Disk Type**, and **VM Type** fields, choose **Automatic** from the drop-down menu to allocate the recommended resources for the job. If the **Persistent Disk Type** field reads **None**, the job does not require persistent disk space.

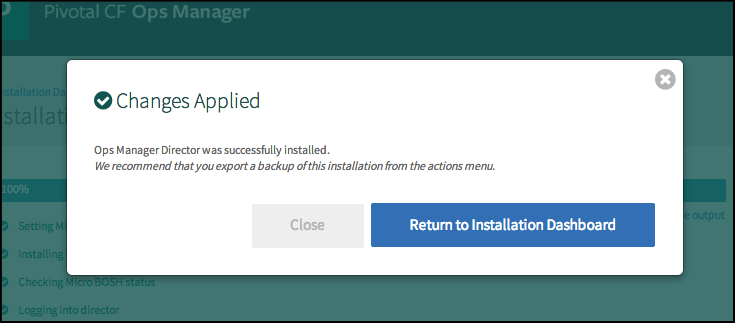
**Note**: If you set a field to **Automatic** and the recommended resource allocation changes in a future version, Ops Manager automatically uses the updated recommended allocation.

Step 10: Complete the Ops Manager Director Installation

1. Click the **Installation Dashboard** link to return to the Installation Dashboard.
2. Click **Apply Changes**. If the following ICMP error message appears, click **Ignore errors and start the install**.



1. Ops Manager Director installs. This may take a few moments. When the installation process successfully completes, the **Changes Applied** window appears.

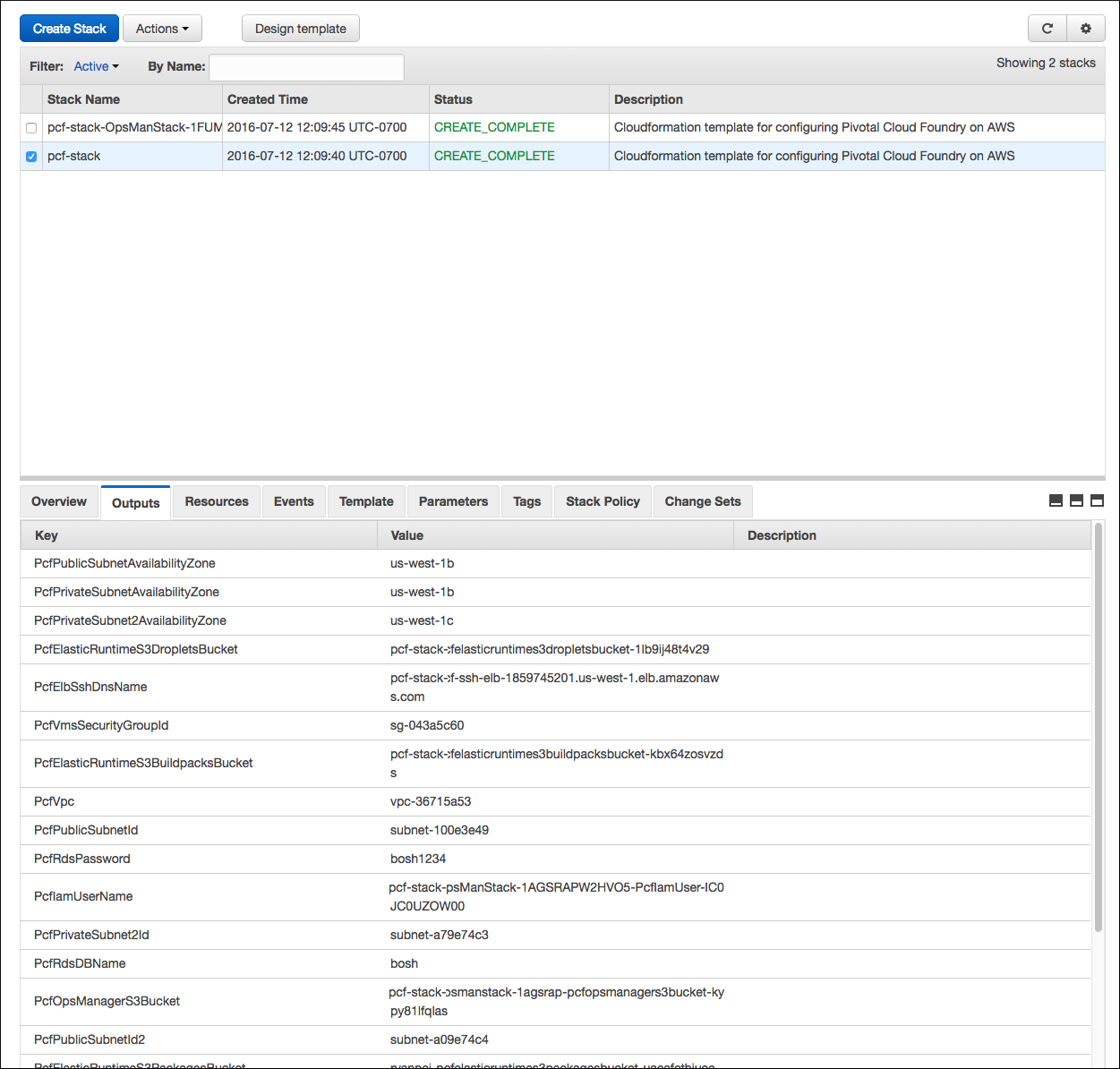


# Deploying Elastic Runtime on AWS v1.10

* [Step 1: Open the Outputs Tab in AWS](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#open-outputs)
* [Step 2: Add Elastic Runtime to Ops Manager](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#add-er)
* [Step 3: Assign Availability Zones and Networks](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#assign-az)
* [Step 4: Add CNAME Record for Your Custom Domain](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#cname)
* [Step 5: Configure Domains](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#er-domain-config)
* [Step 6: Configure Networking](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#networking)
* [Step 7: Configure Application Containers](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#application-containers-config)
* [Step 8: Configure Application Developer Controls](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#er-appdevctrl-config)
* [Step 9: Review Application Security Groups](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#app-security)
* [Step 10: Configure Authentication and Enterprise SSO](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#er-auth-config)
* [Step 11: Configure System Databases](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#sys-db)
  + [Internal Database Configuration](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#internal-db)
  + [Create External System Databases](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#create-dbs)
* [Step 12: (Optional) Configure Internal MySQL](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#internal-mysql)
* [Step 13: Configure File Storage](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#filestore)
  + [Internal Filestore](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#internal_filestore)
  + [External S3 or Ceph Filestore](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#external_s3)
  + [Other IaaS Storage Options](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#other)
* [Step 14: (Optional) Configure System Logging](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#sys-logging)
* [Step 15: (Optional) Customize Apps Manager](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#customize-apps-man)
* [Step 16: (Optional) Configure Email Notifications](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#smtp)
* [Step 17: (Optional) Add CCDB Restore Key](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#ccdb-key)
* [Step 18: Configure Smoke Tests](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#config-smoke-test)
* [Step 19: (Optional) Enable Advanced Features](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#advanced-features)
* [Step 20: Configure Errands](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#errands)
* [Step 21: Configure Router to Elastic Load Balancer](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#config-elb)
* [Step 22: (Optional) Disable Unused Resources](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#disable-resources)
* [Step 23: Download Stemcell](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#stemcell)
* [Step 24: Complete the Elastic Runtime Installation](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#complete)

**Step 1: Open the Outputs Tab in AWS**

1. In the dashboard of your [AWS Console](https://console.aws.amazon.com/), click **CloudFormation**. The Stacks Dashboard displays.
2. Select the **pcf-stack** checkbox, then select the **Outputs** tab.

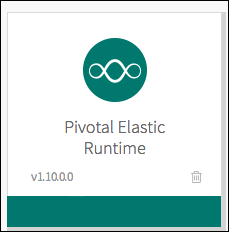


In the steps described below, use the information from the **Value** column of the **Outputs** tab to configure your PCF installation.

## **Step 2: Add Elastic Runtime to Ops Manager**

1. Navigate to the Pivotal Cloud Foundry Operations Manager Installation Dashboard.
2. If you have not downloaded Elastic Runtime, click the Pivotal Network link on the left to download the Elastic Runtime .pivotal file.

Click **Import a Product** to add the tile to Ops Manager. Click the **Elastic Runtime** tile in the Installation Dashboard.



## **Step 3: Assign Availability Zones and Networks**

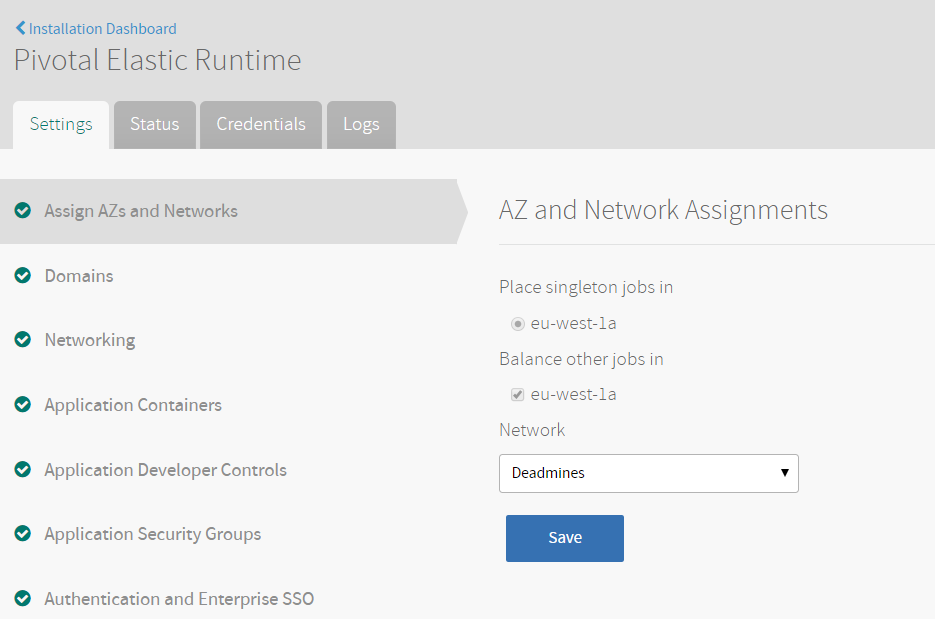
1. Select **Assign AZ and Networks**.
2. Select an Availability Zone under **Place singleton jobs**.

Ops Manager runs any job with a single instance in this Availability Zone.

1. Select one or more Availability Zones under **Balance other jobs**.

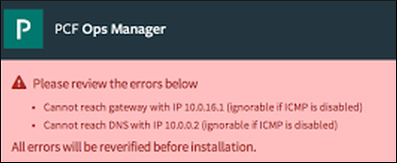
Ops Manager balances instances of jobs with more than one instance across the Availability Zones that you specify.

1. From the **Network** drop-down box, choose the network on which you want to run Elastic Runtime.



1. Click **Save**.

**Note**: When you save this form, a verification error displays because the PCF security group blocks ICMP. You can ignore this error.

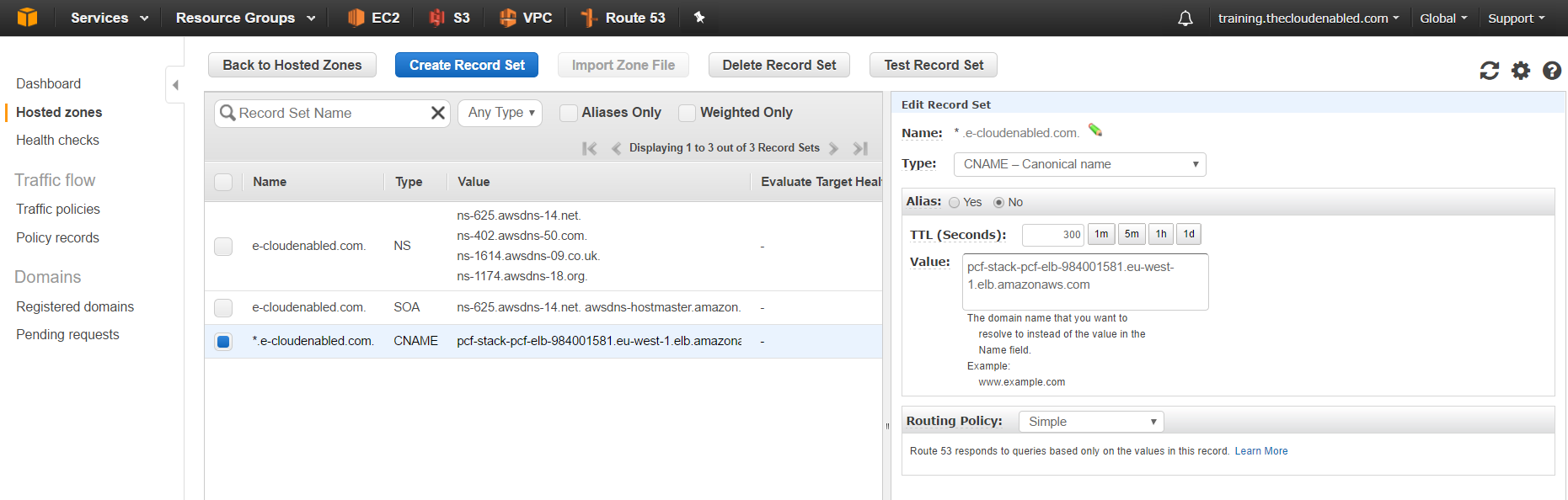


## **Step 4: Add CNAME Record for Your Custom Domain**

In the [Use the AWS CLI to upload your SSL Cert](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-template.html#upload-cert) step, you uploaded an SSL certificate for your PCF wildcard domain to AWS. In this step you redirect all wildcard queries for your domain to the DNS name of your ELB.

**Note**: Do not point your wildcard domain at the numeric IP address for your ELB because this changes frequently.

1. Find the DNS hostname of your ELB. The **Output** tab of the CloudFormation page in the AWS dashboard lists this as the value for the key **PcfElbDnsName**.
2. Log in to the DNS registrar that hosts your domain (for example, AWS R53).
3. Create a CNAME record with your DNS registrar that points \*.YOUR-DOMAIN.com to the DNS hostname of your ELB.
4. Save changes within the web interface of your DNS registrar.



1. In the terminal, run the following dig command to confirm that you created your CNAME record successfully:

dig xyz.MY-DOMAIN.COM

You should see the CNAME record that you just created:

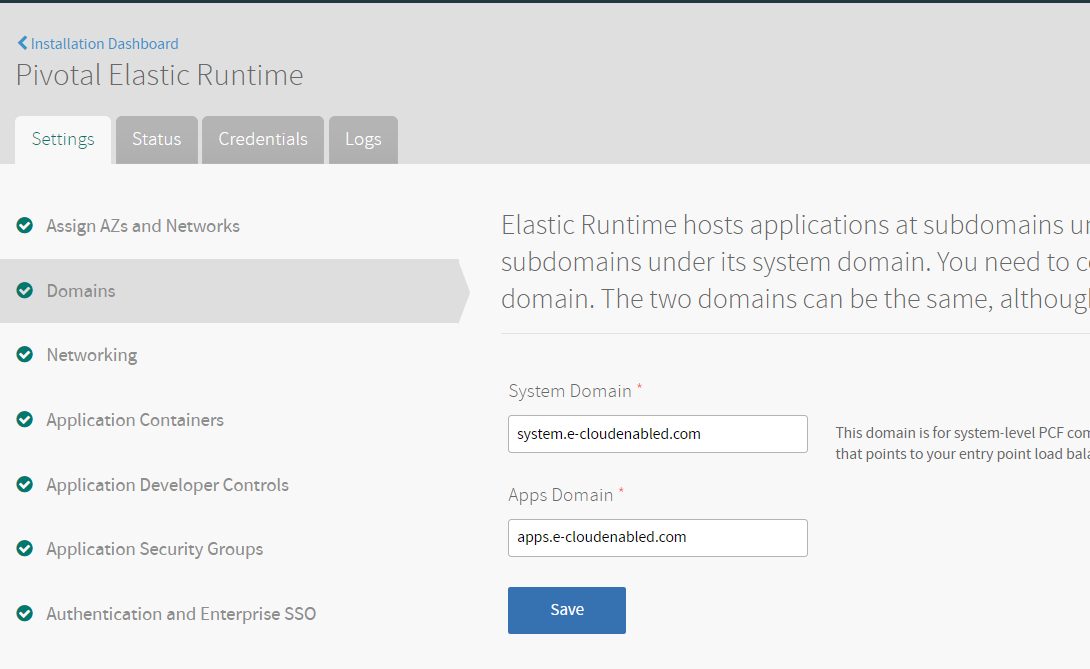
;; ANSWER SECTION:

xyz.MY-DOMAIN.COM. 1767 IN CNAME pcf-ert-frankfurt-pcf-elb-428333773.eu-central-1.elb.amazonaws.com.

**Note**: You **must** complete this step before proceeding to Cloud Controller configuration. A problem that is difficult to resolve can occur if the wildcard domain is improperly cached before the CNAME is registered.

## **Step 5: Configure Domains**

1. Select **Domains**.



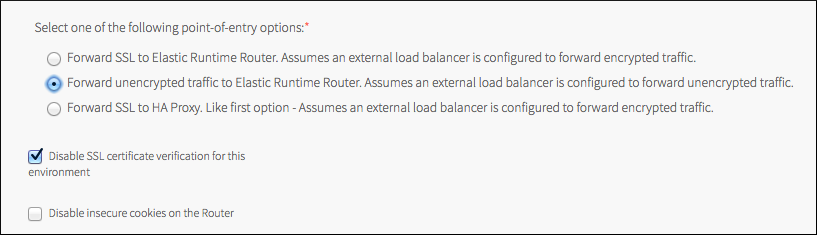
1. Enter the system and application domains.
   * The **System Domain** defines your target when you push apps to Elastic Runtime.
   * The **Apps Domain** defines where Elastic Runtime should serve your apps.

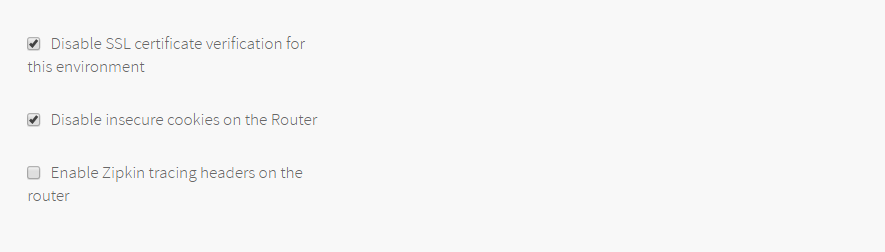
**Note**: Pivotal recommends that you use the same domain name but different subdomain names for your system and app domains. This prevents system and apps routes from overlapping. You will require two wildcard DNS entries: one for the system and the other for apps. For example, \*.system.EXAMPLE.COM and \*.apps.EXAMPLE.COM. Point both wildcard domains at your internal router IP address, which can be found under the status tab in the Elastic Runtime tile.

**Note**: You configured a wildcard DNS record for these domains in an earlier step.

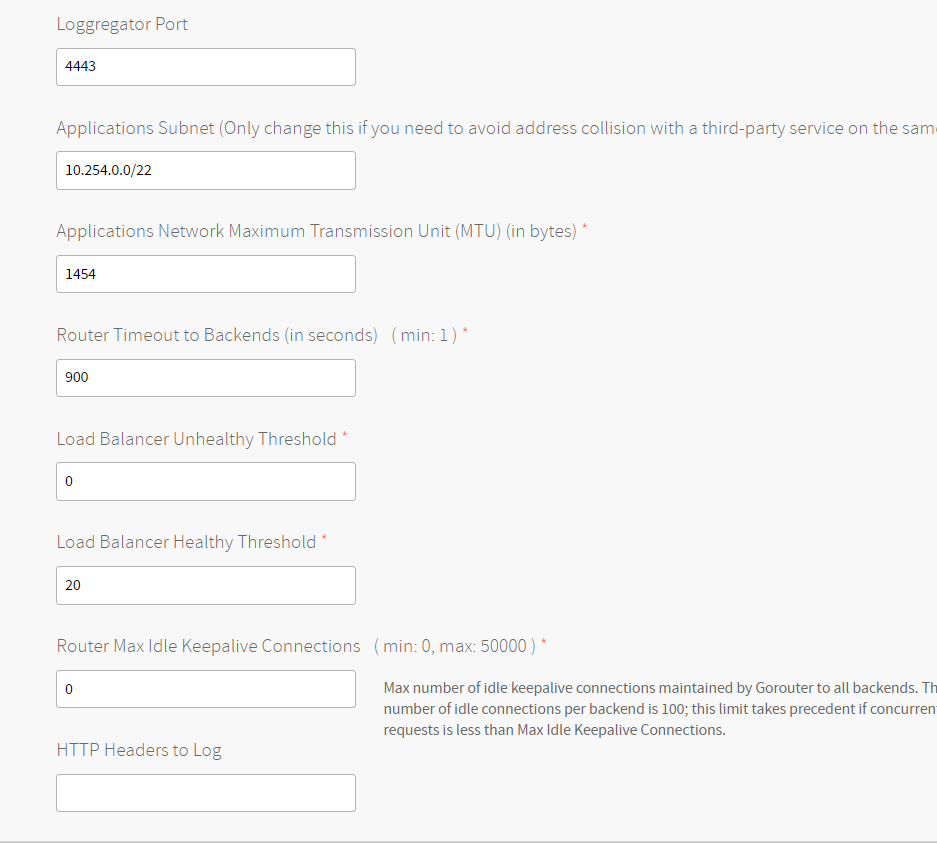
1. Click **Save**.

## **Step 6: Configure Networking**

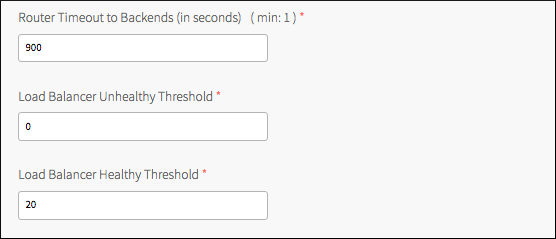
1. Select **Networking**.
2. Leave the **Router IPs**, **SSH Proxy IPs**, **HAProxy IPs**, and **TCP Router IPs** fields blank. You do not need to complete these fields when deploying PCF to AWS with Elastic Load Balancers (ELBs).
3. Under **Configure the point-of-entry to this environment**, choose one of the following options:
   * **Forward unencrypted traffic to Elastic Runtime Router**: Select this option if your deployment uses an external load balancer that cannot forward encrypted traffic to the Elastic Runtime Router, or for a development environment that does not require load balancing. 
4. If you are not using SSL encryption or if you are using self-signed certificates, select **Disable SSL certificate verification for this environment**. Selecting this checkbox also disables SSL verification for route services.
5. Select the **Disable insecure cookies on the Router** checkbox to set the secure flag for cookies generated by the router.
6. To disable the addition of Zipkin tracing headers on the Gorouter, deselect the **Enable Zipkin tracing headers on the router** checkbox

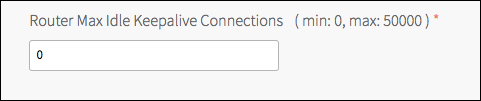


1. In the **Choose whether or not to enable route services** section, choose **Enable route services**
2. For Loggregator Port, you must enter 4443. In AWS deployments, port 4443 forwards SSL traffic that supports WebSockets from the ELB. Do not use the default port of 443.
3. (Optional) Use the **Applications Subnet** field if you need to avoid address collision with a third-party service on the same subnet as your apps. Enter a CIDR subnet mask specifying the range of available IP addresses assigned to your app containers. The IP range must be different from the network used by the system VMs.
4. (Optional) You can change the value in the **Applications Network Maximum Transmission Unit (MTU)** field. Pivotal recommends setting the MTU value for your application network to 1454. Some configurations, such as networks that use GRE tunnels, may require a smaller MTU value.



1. (Optional) Increase the value of **Load Balancer Unhealthy Threshold** to specify the amount of time, in seconds, that the router continues to accept connections before shutting down. During this period, healthchecks may report the router as unhealthy, which causes load balancers to failover to other routers. Set this value to an amount greater than or equal to the maximum time it takes your load balancer to consider a router instance unhealthy, given contiguous failed healthchecks.
2. (Optional) Modify the value of **Load Balancer Healthy Threshold**. This field specifies the amount of time, in seconds, to wait until declaring the Router instance started. This allows an external load balancer time to register the Router instance as healthy.



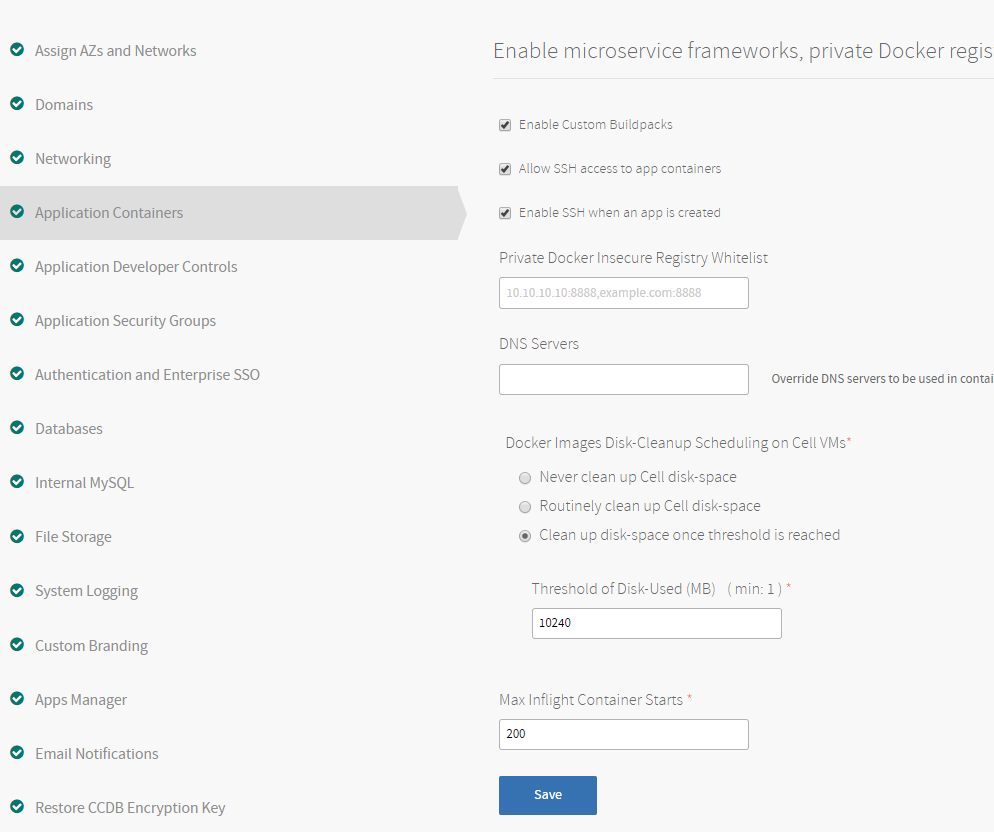
1. Enter a value for **Router Max Idle Keepalive Connections**. 
2. (Optional) If app developers in your organization want certain HTTP headers to appear in their app logs with information from the Gorouter, specify them in the **HTTP Headers to Log** field. 
3. TCP Routing is disabled by default. To enable this feature, perform the following steps:

Select the **Enable TCP Routing** radio button.

1. Click **Save**.

## **Step 7: Configure Application Containers**

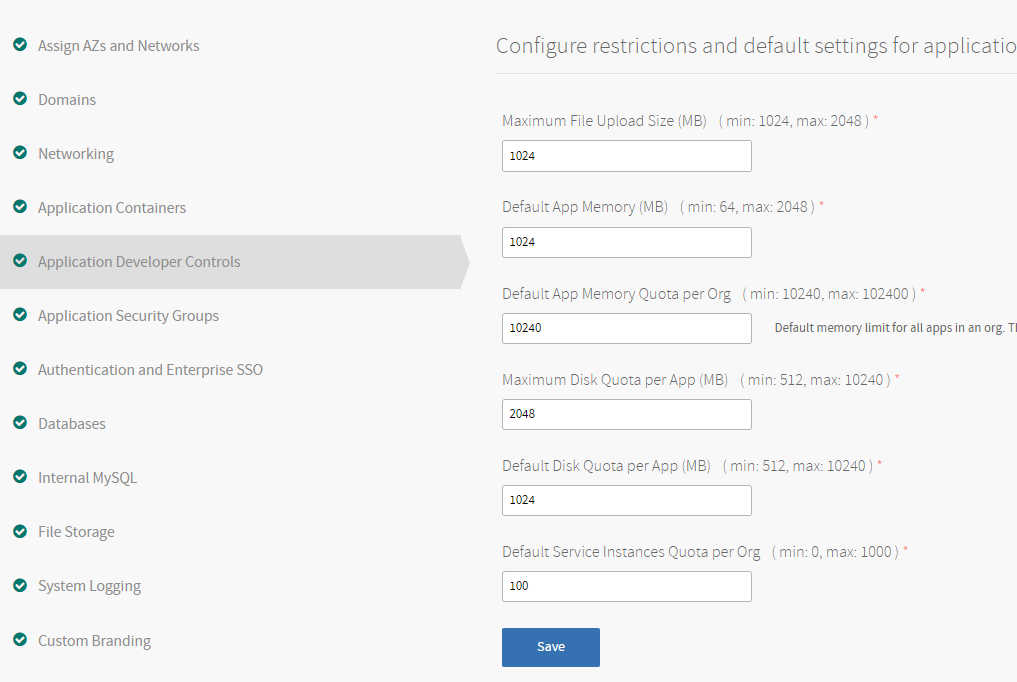
1. Select **Application Containers**.



1. The **Enable Custom Buildpacks** checkbox governs the ability to pass a custom buildpack URL to the -b option of the cf push command. By default, this ability is enabled, letting developers use custom buildpacks when deploying apps. Disable this option by disabling the checkbox.
2. The **Allow SSH access to app containers** checkbox controls SSH access to application instances. Enable the checkbox to permit SSH access across your deployment, and disable it to prevent all SSH access.
3. If you want enable SSH access for new apps by default in spaces that allow SSH, select **Enable SSH when an app is created**. If you deselect the checkbox, developers can still enable SSH after pushing their apps by running cf enable-ssh APP-NAME.
4. You can configure Elastic Runtime to run app instances in Docker containers by supplying their IP address range(s) in the **Private Docker Insecure Registry Whitelist** textbox.
5. By default, containers use the same DNS servers as the host. If you want to override the DNS servers to be used in containers, enter a comma-separated list of servers in **DNS Servers**.
6. Select your preference for **Docker Images Disk-Cleanup Scheduling on Cell VMs**. If you choose **Clean up disk-space once threshold is reached**, enter a **Threshold of Disk-Used** in megabytes.
7. Optionally, enter a number in the **Max Inflight Container Starts** textbox. This number configures the maximum number of started instances across your deployment’s Diego Cells.
8. Click **Save**.

## **Step 8: Configure Application Developer Controls**

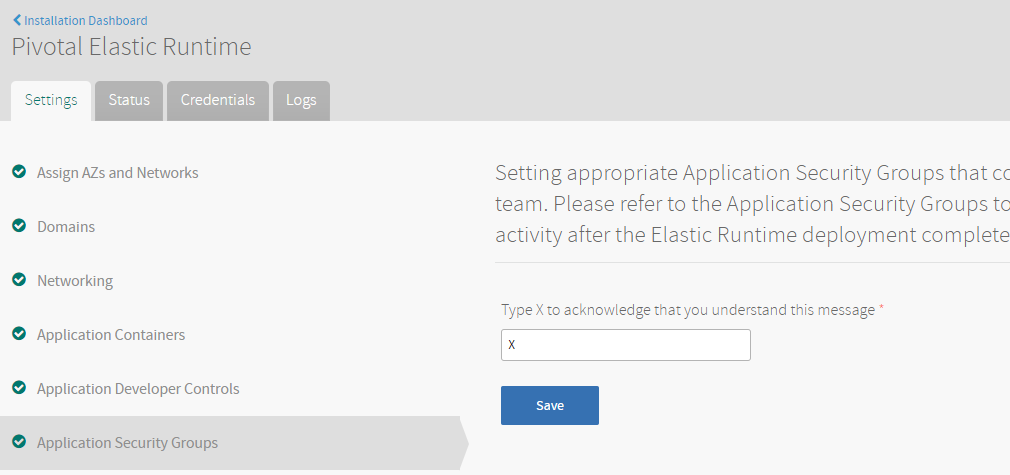
1. Select **Application Developer Controls**.



1. Enter your intended maximum file upload size.
2. Enter your default RAM memory allocation per app.
3. Enter your default total RAM memory (RAM) quota per Org. You can change this in the CLI.
4. Enter your maximum and default disk quotas per app.
5. Enter your default service instances quota per Org. You can change this in the CLI.
6. Click **Save**.

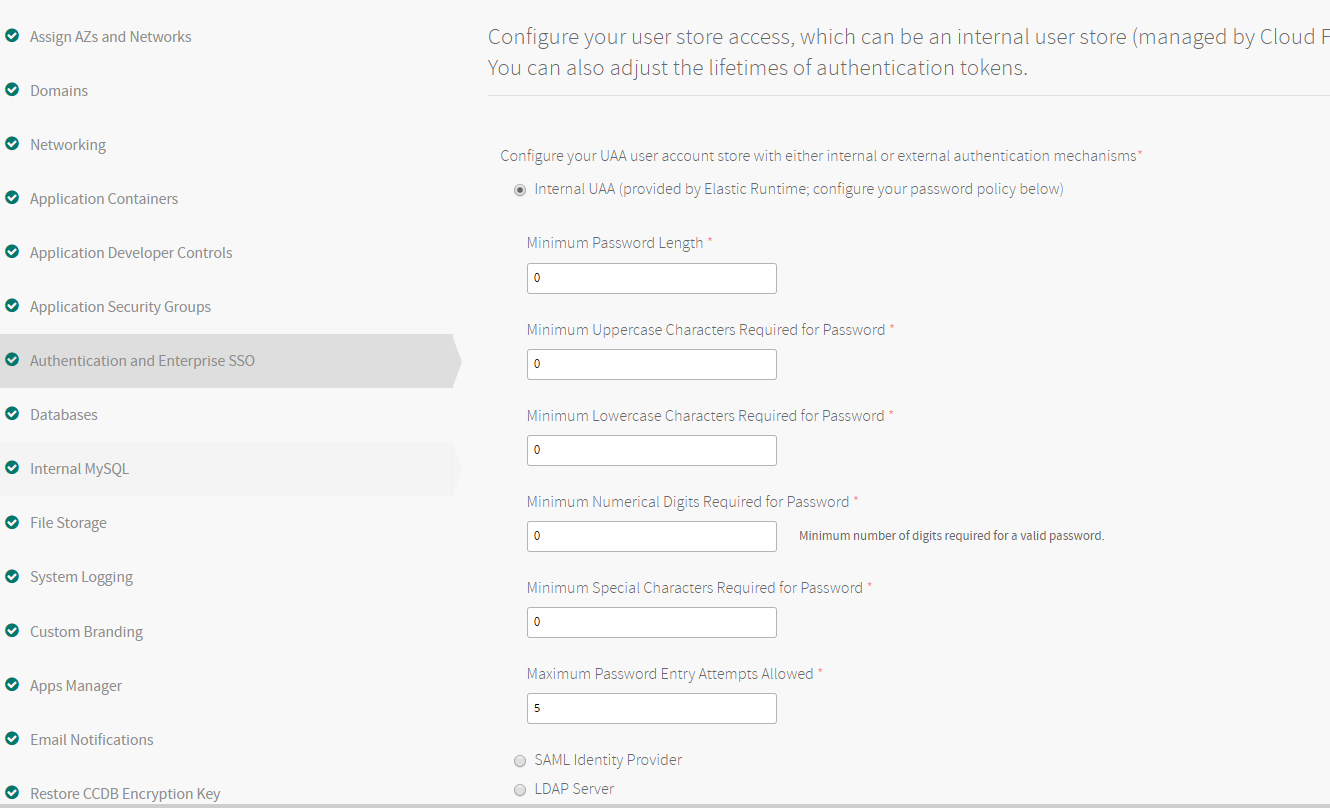
## **Step 9: Review Application Security Groups**

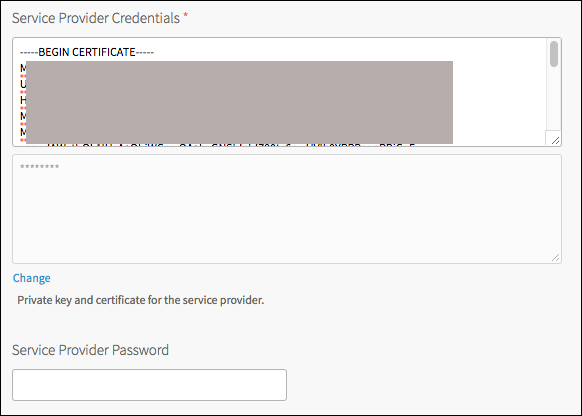
Setting appropriate [Application Security Groups](https://docs.pivotal.io/pivotalcf/1-10/concepts/asg.html) is critical for a secure deployment. Type X in the box to acknowledge that once the Elastic Runtime deployment completes, you will review and set the appropriate application security groups.

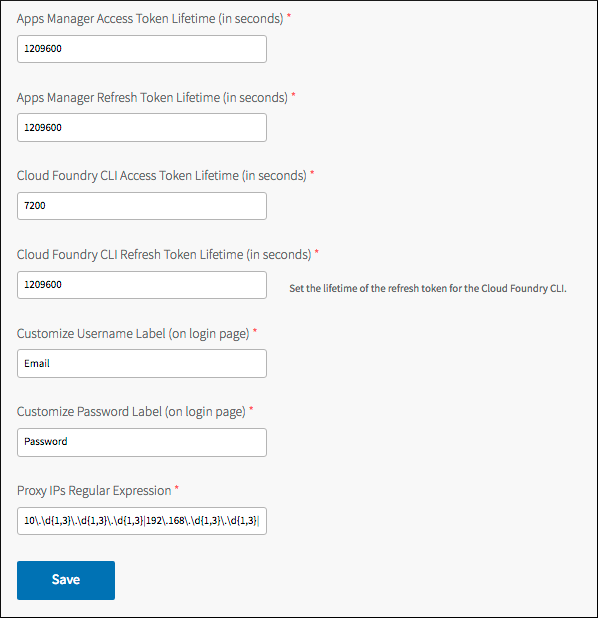


## **Step 10: Configure Authentication and Enterprise SSO**

1. Select **Authentication and Enterprise SSO**.



1. To authenticate user sign-ons, your deployment can use one of three types of user database: the UAA server’s internal user store, an external SAML identity provider, or an external LDAP server.
   * To use the internal UAA, select the **Internal** option.
2. Under **Service Provider Credentials**, enter a certificate and private key to be used by UAA as a SAML Service Provider for signing outgoing SAML authentication requests. You can generate a self-signed certificate. The following domains must be associated with the certificate: login.YOUR-SYSTEM-DOMAIN or \*.login.YOUR-SYSTEM-DOMAIN.
3. If the private key specified under **Service Provider Credentials** is password-protected, enter the password under **Service Provider Passwor****d**.
4. (Optional) Lets the defaults to be set for **Apps Manager Access Token Lifetime**, **Apps Manager Refresh Token Lifetime**, **Cloud Foundry CLI Access Token Lifetime**, and **Cloud Foundry CLI Refresh Token Lifetime** fields, change the lifetimes of tokens granted for Apps Manager and Cloud Foundry Command Line Interface (cf CLI) login access and refresh. Most deployments use the defaults.
5. (Use Default) Customize the text prompts used for username and password from the cf CLI and Apps Manager login popup.
6. (Use Default l) The **Proxy IPs Regular Expression** field contains a pipe-delimited set of regular expressions that UAA considers to be reverse proxy IP addresses. UAA respects the x-forwarded-for and x-forwarded-proto headers coming from IP addresses that match these regular expressions. To configure UAA to respond properly to Router or HAProxy requests coming from a public IP address, append a regular expression or regular expressions to match the public IP address.



1. Click **Save**.

## **Step 11: Configure System Databases**

You can configure Elastic Runtime to use the internal MySQL database provided with PCF, or you can configure an external database provider for the databases required by Elastic Runtime.

1. Select External System Databases

If you want to use an external database provider for your Elastic Runtime databases, you must first create the databases on the RDS instance provided by the CloudFormation script.

To create the required databases on an AWS RDS instance, perform the following steps.

1. Add the AWS-provided key pair to your SSH profile so that you can access the Ops Manager VM:

ssh-add aws-keypair.pem

Or login through putty to Ops manager VM by using aws key pair.

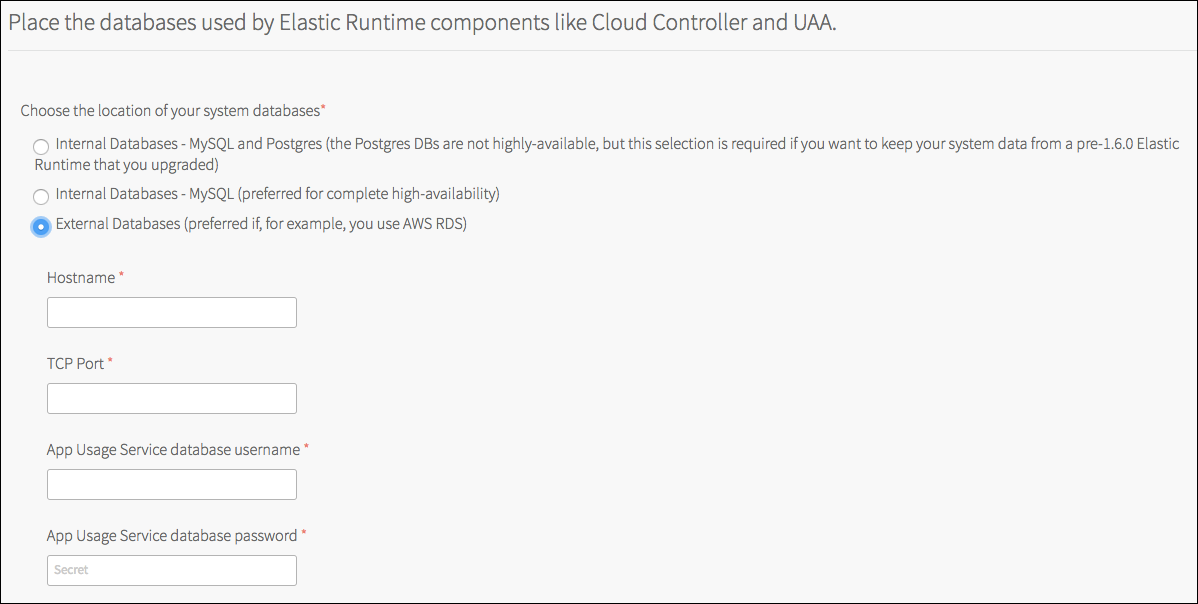
1. SSH in to your Ops Manager using the [Ops Manager FQDN](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-om-deploy.html#create-dns) and the username ubuntu:

ssh ubuntu@OPS\_MANAGER\_FQDN

1. Run the following terminal command to log in to your RDS instance through the MySQL client, using values from your AWS dashboard [Outputs tab](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#open-outputs) to fill in the following keys:

mysql --host=PcfRdsAddress --user=PcfRdsUsername --password=PcfRdsPassword

1. Run the following MySQL commands to create databases for the seven Elastic Runtime components that require a relational database:
2. CREATE database uaa;
3. CREATE database ccdb;
4. CREATE database notifications;
5. CREATE database autoscale;
6. CREATE database app\_usage\_service;
7. CREATE database routing;
8. CREATE database diego;
9. CREATE database account;
10. CREATE database nfsvolume;
11. CREATE database networkpolicyserver;
12. Type exit to quit the MySQL client, and exit again to close your connection to the Ops Manager VM.
13. In Elastic Runtime, select **Databases**.
14. Select the **External Databases** option.



1. For the **Hostname** and **TCP Port** fields, enter the corresponding values from the [Outputs tab](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#open-outputs) in the AWS Console, according to the following table:

|  |  |
| --- | --- |
| **Elastic Runtime Field** | **Outputs Key** |
| Hostname | PcfRdsAddress |
| TCP Port | PcfRdsPort |

1. For each **database username** and **database password** field, enter the corresponding values from the [Outputs tab](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#open-outputs) in the AWS Console, according to the following table:

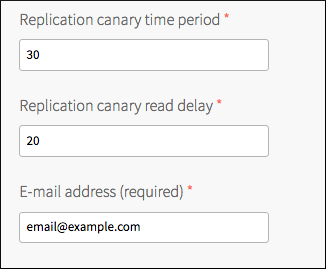
|  |  |
| --- | --- |
| **Elastic Runtime Field** | **Outputs Key** |
| DATABASE-NAME database username | PcfRdsUsername |
| DATABASE-NAME database password | PcfRdsPassword |

1. Click **Save**.

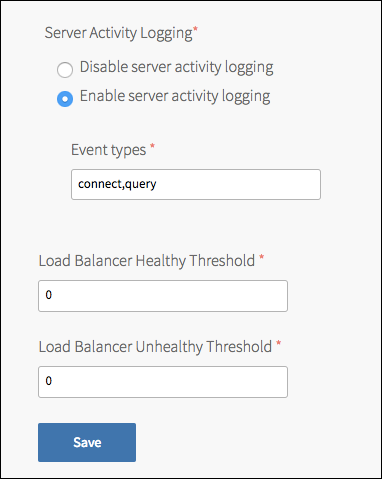
## **Step 12: (Optional) Configure Internal MySQL**

**Note**: You only need to configure this section if you have selected **Internal Databases - MySQL** in the **Databases** section.

1. Select **Internal MySQL**.
2. In the **MySQL Proxy IPs** field, leave blank
3. For **MySQL Service Hostname**, leave blank
4. In the **Replication canary time period** field, leave the default of 30 seconds or modify the value based on the needs of your deployment.
5. In the **Replication canary read delay** field, leave the default of 20 seconds or modify the value based on the needs of your deployment..
6. (**Required**): In the **E-mail address** field, enter the email address where the MySQL service sends alerts when the cluster experiences a replication issue or when a node is not allowed to auto-rejoin the cluster.



1. Under **Automated Backups Configuration**, choose one of three options for MySQL backups:
   * **Disable automatic backups of MySQL**
2. If you want to log audit events for internal MySQL, select **Enable server activity logging** under **Server Activity Logging**.
   * For the **Event types** field, you can enter the events you want the MySQL service to log. By default, this field includes connect and query, which tracks who connects to the system and what queries are processed..



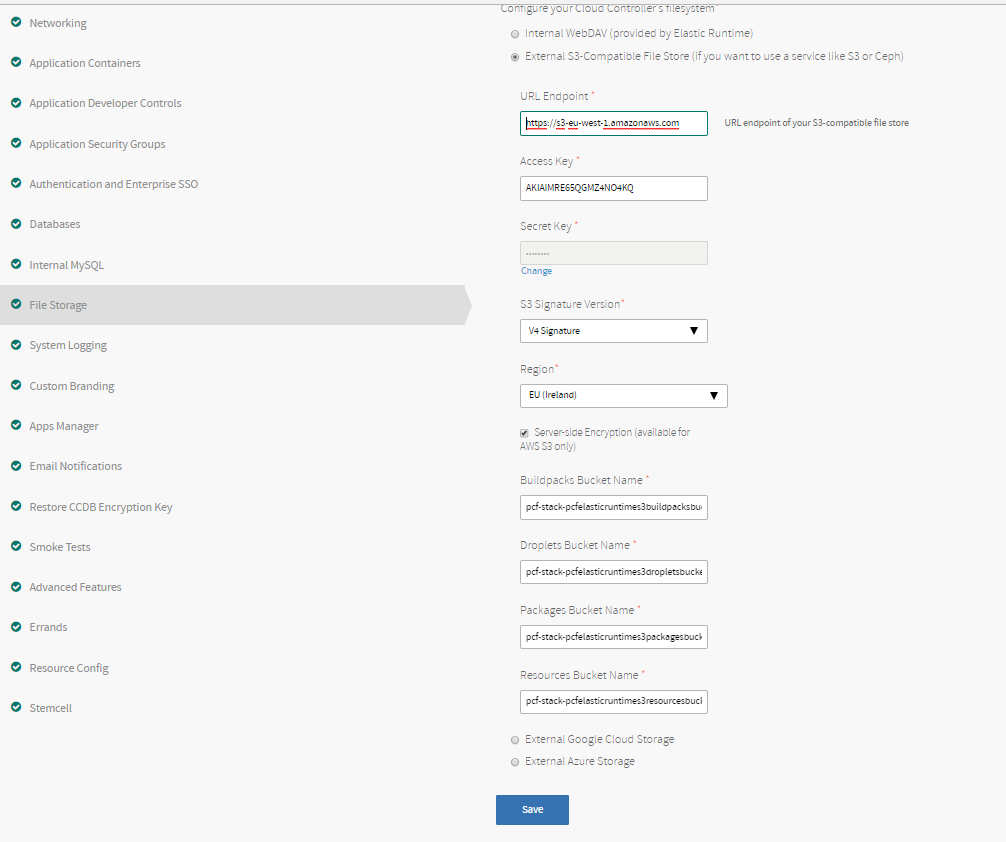
1. Enter values for the following fields:
   * **Load Balancer Healthy Threshold**: Specifies the amount of time, in seconds, to wait until declaring the MySQL proxy instance started. This allows an external load balancer time to register the instance as healthy.
   * **Load Balancer Unhealthy Threshold**: Specifies the amount of time, in seconds, that the MySQL proxy continues to accept connections before shutting down. During this period, the healthcheck reports as unhealthy to cause load balancers to fail over to other proxies. You must enter a value greater than or equal to the maximum time it takes your load balancer to consider a proxy instance unhealthy, given repeated failed healthchecks.
2. Click **Save**.

## **Step 13: Configure File Storage**

To minimize system downtime, Pivotal recommends using highly resilient and redundant externalfilestores for your Elastic Runtime file storage.

When configuring file storage for the Cloud Controller in Elastic Runtime, you select one of the following:

* External S3-compatible



1. In the Elastic Runtime tile, select **File Storage**.
2. Select the **External S3-Compatible Filestore** option and complete the following fields:
   * For **URL Endpoint**:
     1. In a browser, open the [Amazon Simple Storage Service (Amazon S3) table](http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region).
     2. Prepend https:// to the **Endpoint** for your region and copy it into the Ops Manager **URL Endpoint** field.   
        For example, in the **us-west-2** region, use https://s3-us-west-2.amazonaws.com/.
   * For **S3 Signature Version** and **Region**, use the **V4 Signature** values. AWS recommends using [Signature Version 4](http://docs.aws.amazon.com/general/latest/gr/signing_aws_api_requests.html#signature-versions).
   * Select **Server-side Encryption (available for AWS S3 only)** to encrypt the contents of your S3 filestore. See the [AWS S3 documentation](http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html) for more information.
   * Use the values in your AWS [Outputs tab](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#open-outputs) to complete the remaining fields as follows:

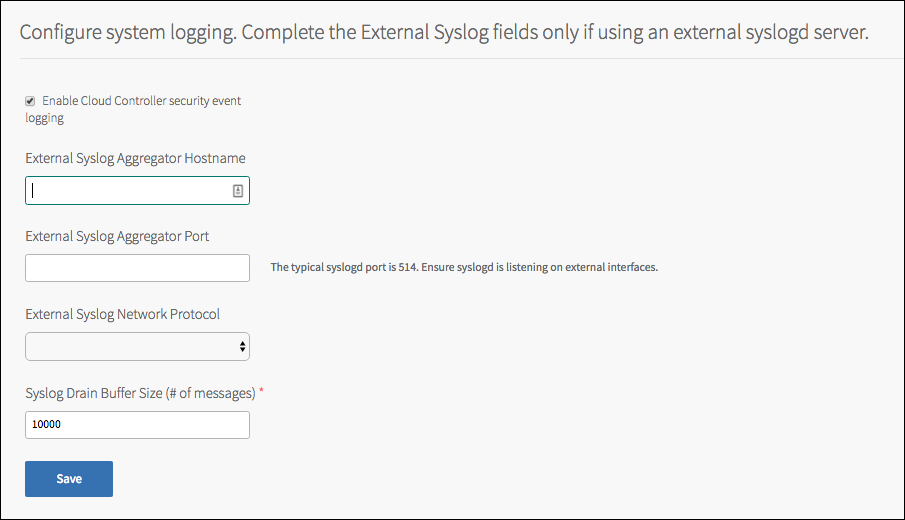
|  |  |
| --- | --- |
| **Ops Manager Field** | **Outputs Key** |
| Buildpacks Bucket Name | PcfElasticRuntimeS3BuildpacksBucket |
| Droplets Bucket Name | PcfElasticRuntimeS3DropletsBucket |
| Packages Bucket Name | PcfElasticRuntimeS3PackagesBucket |
| Resources Bucket Name | PcfElasticRuntimeS3ResourcesBucket |
| Access Key ID | PcfIamUserAccessKey |
| AWS Secret Key | PcfIamUserSecretAccessKey |

1. Click **Save**.

## **Step 14: (Optional) Configure System Logging**

If you forward logging messages to an external Reliable Event Logging Protocol (RELP) server, complete the following steps:

**Note**: For this lap we are not using External Syslog Aggregator

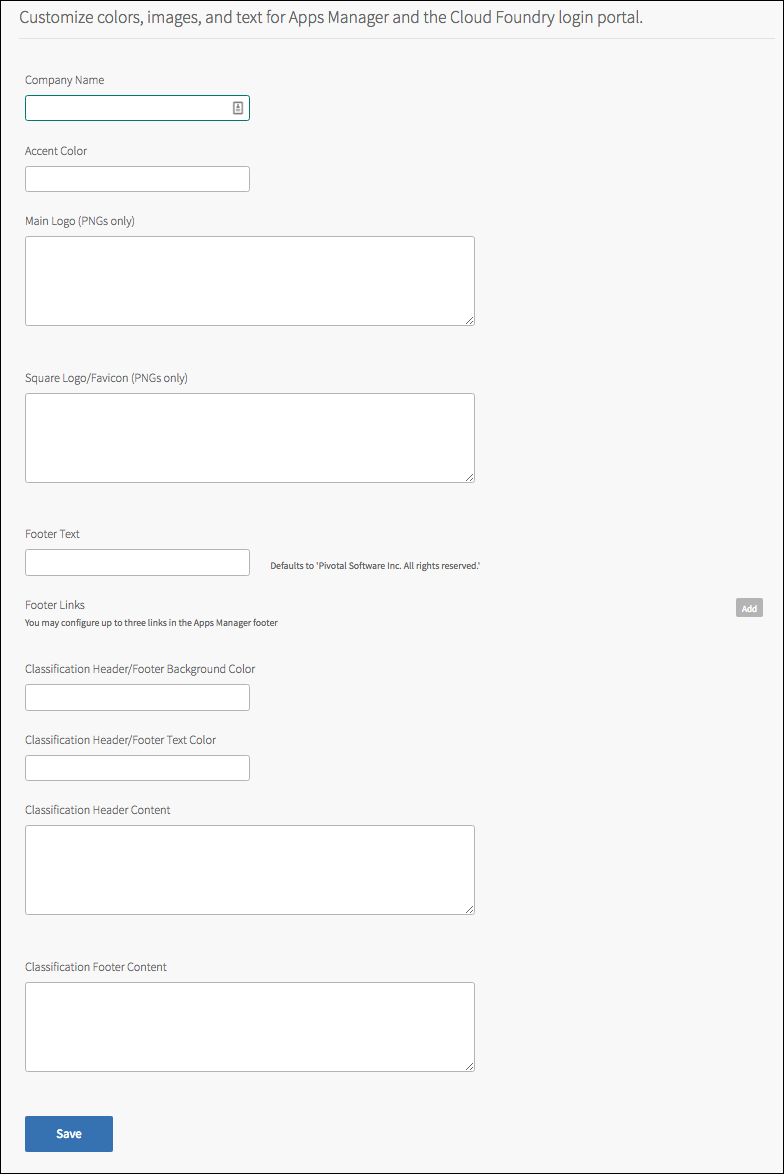
1. Select **System Logging**. 
2. Click **Save**.

## **Step 15: (Optional) Customize Apps Manager**

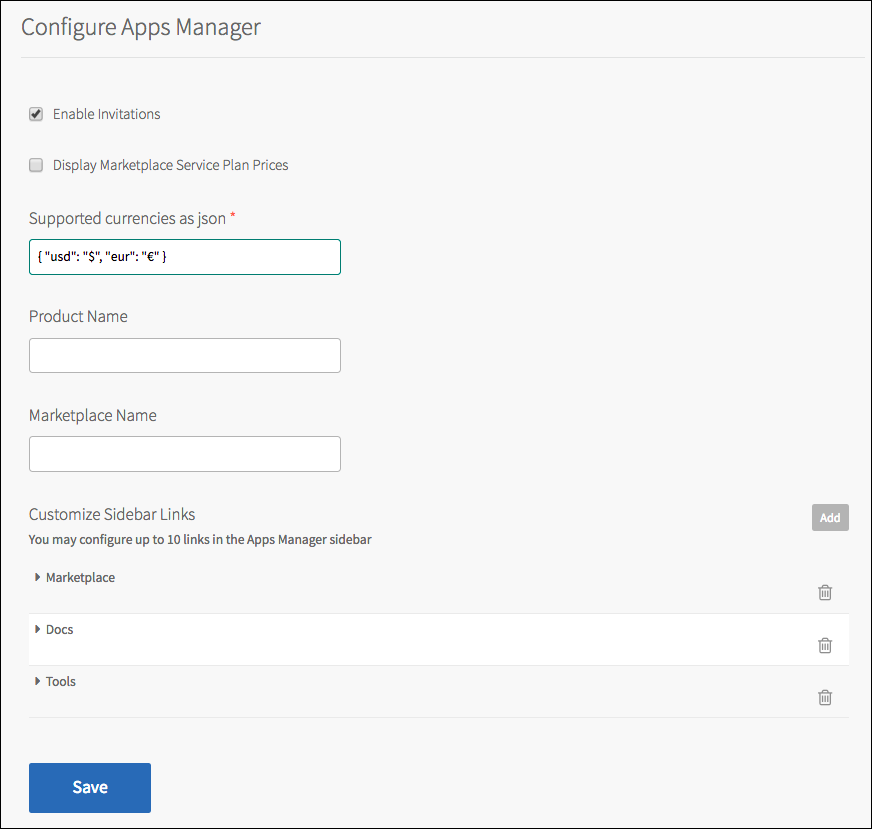
The **Custom Branding** and **Apps Manager** sections customize the appearance and functionality of Apps Manager..

1. Select **Custom Branding**. Use this section to configure the text, colors, and images of the interface that developers see when they log in, create an account, reset their password, or use Apps

Note: Leave the options blank.



1. Click **Save** to save your settings in this section.

Select **Apps Manager**.

1. Select **Enable Invitations** to enable invitations in Apps Manager. Space Managers can invite new users for a given space, Org Managers can invite new users for a given org, and Admins can invite new users across all orgs and spaces..
2. Select **Display Marketplace Service Plan Prices** to display the prices for your services plans in the Marketplace.
3. Enter the **Supported currencies as json** to appear in the Marketplace. This defaults to {"usd": "$", "eur": "€"}.
4. Use **Product Name**, **Marketplace Name**, and **Customize Sidebar Links** to configure page names and sidebar links in the **Apps Manager** and **Marketplace** pages.
5. Click **Save** to save your settings in this section.

## **Step 16: (Optional) Configure Email Notifications**

Elastic Runtime uses SMTP to send invitations and confirmations to Apps Manager users.

1. Select **Email Notifications**.

Note: For this lab , leave this blank

1. Click **Save**.

## **Step 17: (Optional) Add CCDB Restore Key**

Perform this step if all of the following are true:

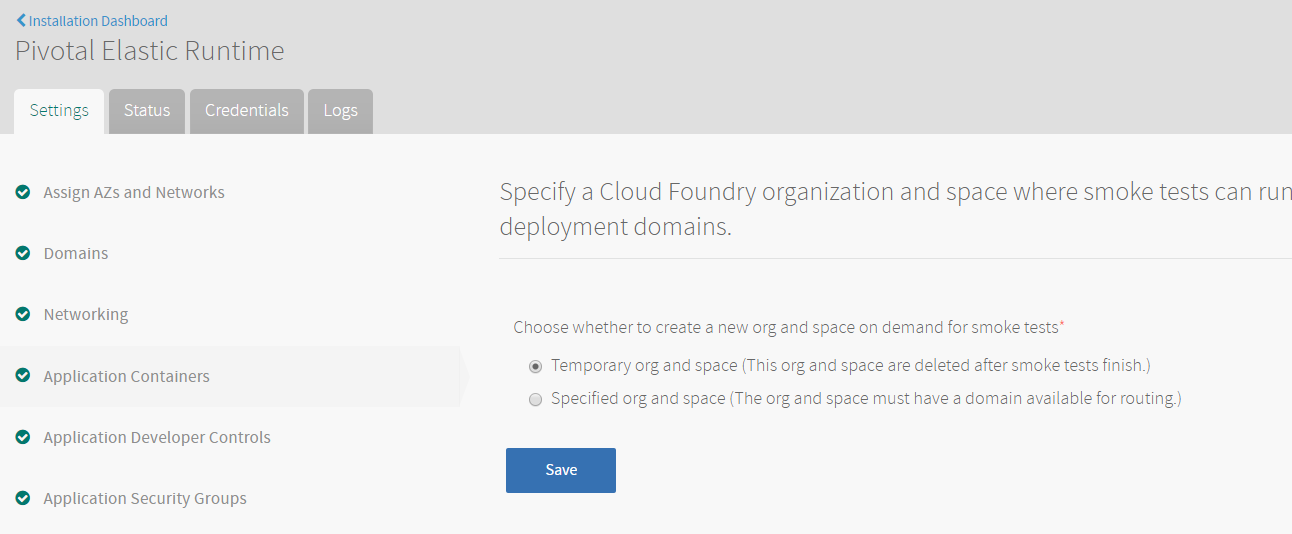
* You deployed Elastic Runtime previously.
* You then stopped Elastic Runtime or it crashed.
* You are re-deploying Elastic Runtime with a backup of your Cloud Controller database.

Note: Leave this blank.

## **Step 18: Configure Smoke Tests**

The Smoke Tests errand runs basic functionality tests against your Elastic Runtime deployment after an installation or update. In this section, choose where to run smoke tests. In the **Errands** section, you can choose whether or not to run the Smoke Tests errand.

1. Select **Smoke Tests**.
2. If you have a shared apps domain, select **Temporary org and space**, which creates an ad-hoc org and space for running smoke tests and deletes them afterwards..



1. Click **Save**.

## **Step 19: (Optional) Enable Advanced Features**

The Advanced Features section of Elastic Runtime includes new functionality that may have certain constraints. Although these features are fully supported, Pivotal recommends caution when using them in production environments.

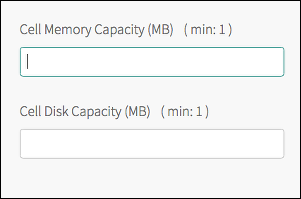
### Diego Cell Memory and Disk Overcommit

If your apps do not use the full allocation of disk space and memory set in the **Resource Config** tab, you might want use this feature. These fields control the amount to overcommit disk and memory resources to each Diego Cell VM.

For example, you might want to use the overcommit if your apps use a small amount of disk and memory capacity compared to the amounts set in the **Resource Config** settings for **Diego Cell**.

To enable overcommit, follow these steps:

1. Select **Advanced Features**.



1. Enter the total desired amount of Diego cell memory value in the **Cell Memory Capacity (MB)** field. Refer to the **Diego Cell** row in the **Resource Config** tab for the current Cell memory capacity settings that this field overrides.
2. Enter the total desired amount of Diego cell disk capacity value in the **Cell Disk Capacity (MB)** field. Refer to the **Diego Cell** row in the **Resource Config** tab for the current Cell disk capacity settings that this field overrides.
3. Whitelist for Non-RFC-1918 Private Networks

Some private networks require extra configuration so that internal file storage (WebDAV) can communicate with other PCF processes.

The **Whitelist for non-RFC-1918 Private Networks** field is provided for deployments that use a non-RFC 1918 private network. This is typically a private network other than 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16.

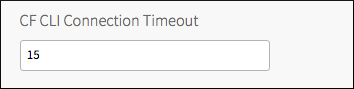
**Used Default: Most PCF deployments do not require any modifications to this field.**

1. CF CLI Connection Timeout

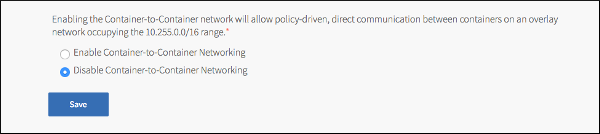
The **CF CLI Connection Timeout** field allows you to override the default five second timeout of the Cloud Foundry Command Line Interface (cf CLI) used within your PCF deployment. This timeout affects the cf CLI command used to push Elastic Runtime errand apps such as Notifications, Autoscaler, and Apps Manager.

Set the value of this field to a higher value, in seconds, if you are experiencing domain name resolution timeouts when pushing errands in Elastic Runtime.

To modify the value of the **CF CLI Connection Timeout**, perform the following steps:

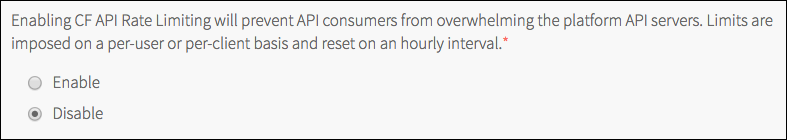
Add a value, in seconds, to the **CF CLI Connection Timeout** field.

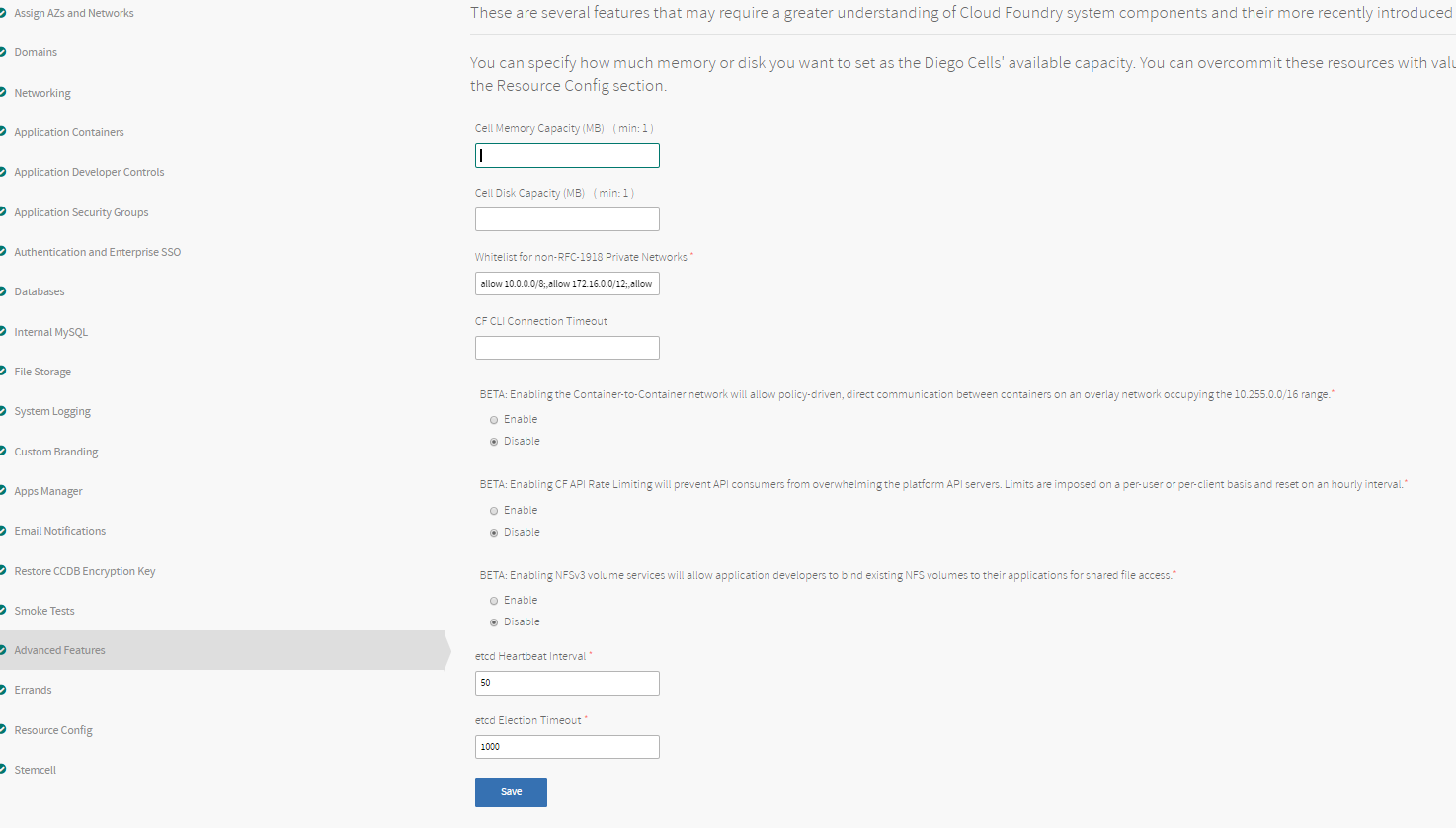
1. Container-to-Container Networking

Select **Disable Container-to-Container Networking**.

1. CF API Rate Limiting

Enabling CF API Rate Limiting prevents API consumers from overwhelming the platform API servers. Limits are imposed on a per-user or per-client basis and reset on an hourly interval.

select **Disable**. 



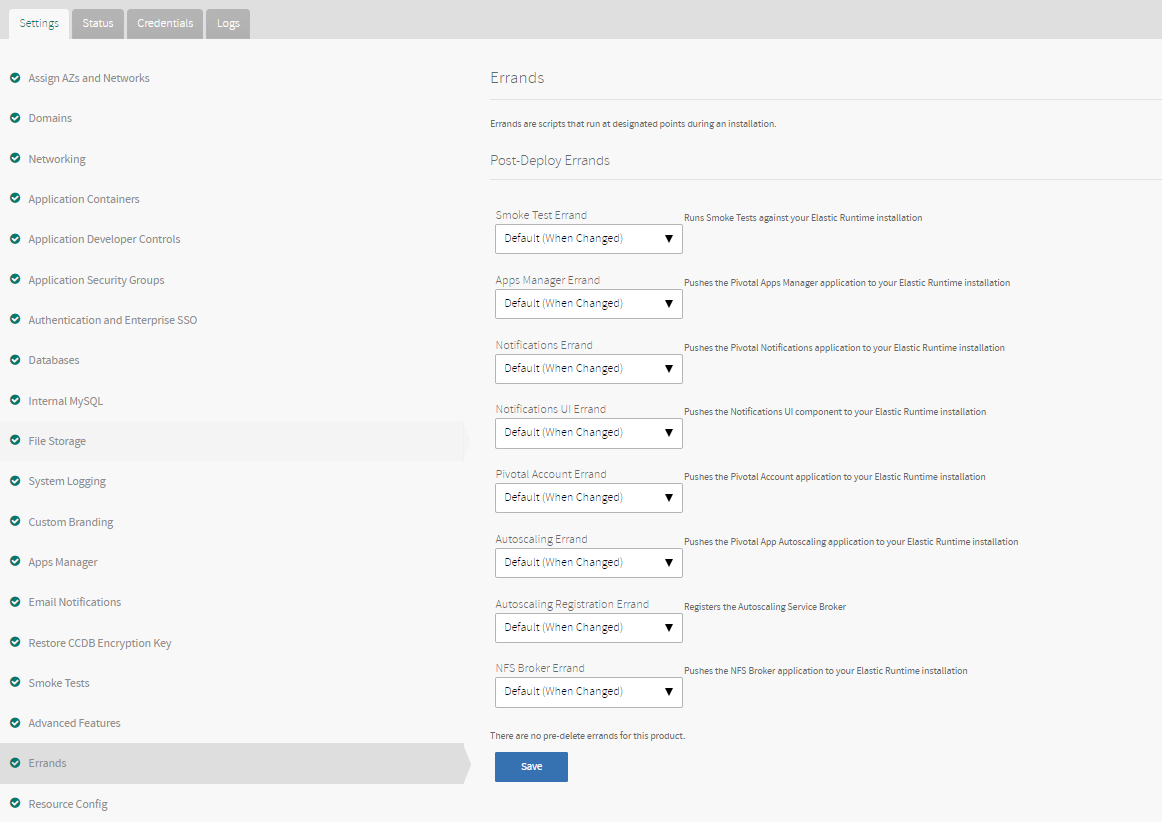
Click **Save**.

## **Step 20: Configure Errands**

Errands are scripts that Ops Manager runs automatically when it installs or uninstalls a product, such as a new version of Elastic Runtime. There are two types of errands: post-deploy errands run after the product is installed, and pre-delete errands run before the product in uninstalled.

By default, Ops Manager always runs pre-delete errands, and only runs post-deploy errands when the product has changed since the last time Ops Manager installed something. The Elastic Runtime tile **Errands** pane lets you change these run rules. For each errand, you can select **On** to run it always, **Off** to never run it, or **When Changed** to run it only when the product has changed since the last install.

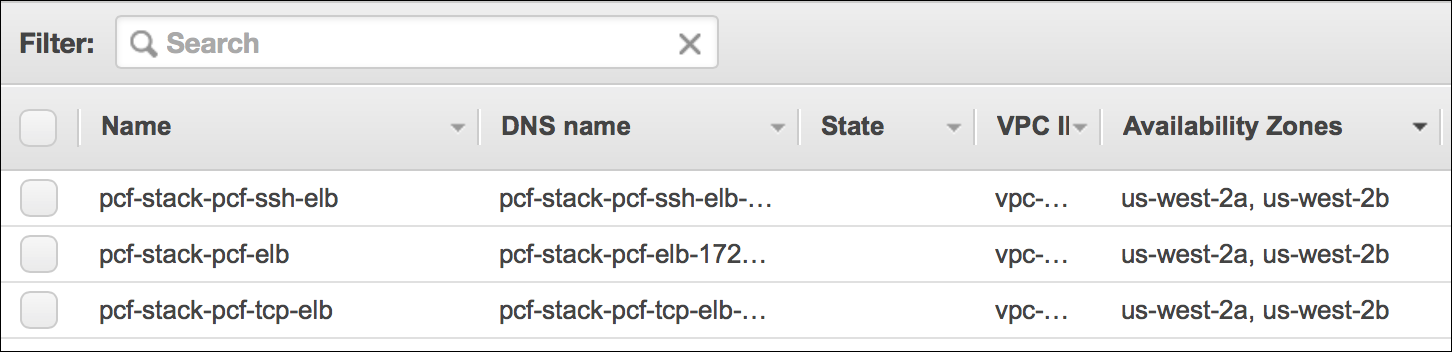
Select **When Changes**

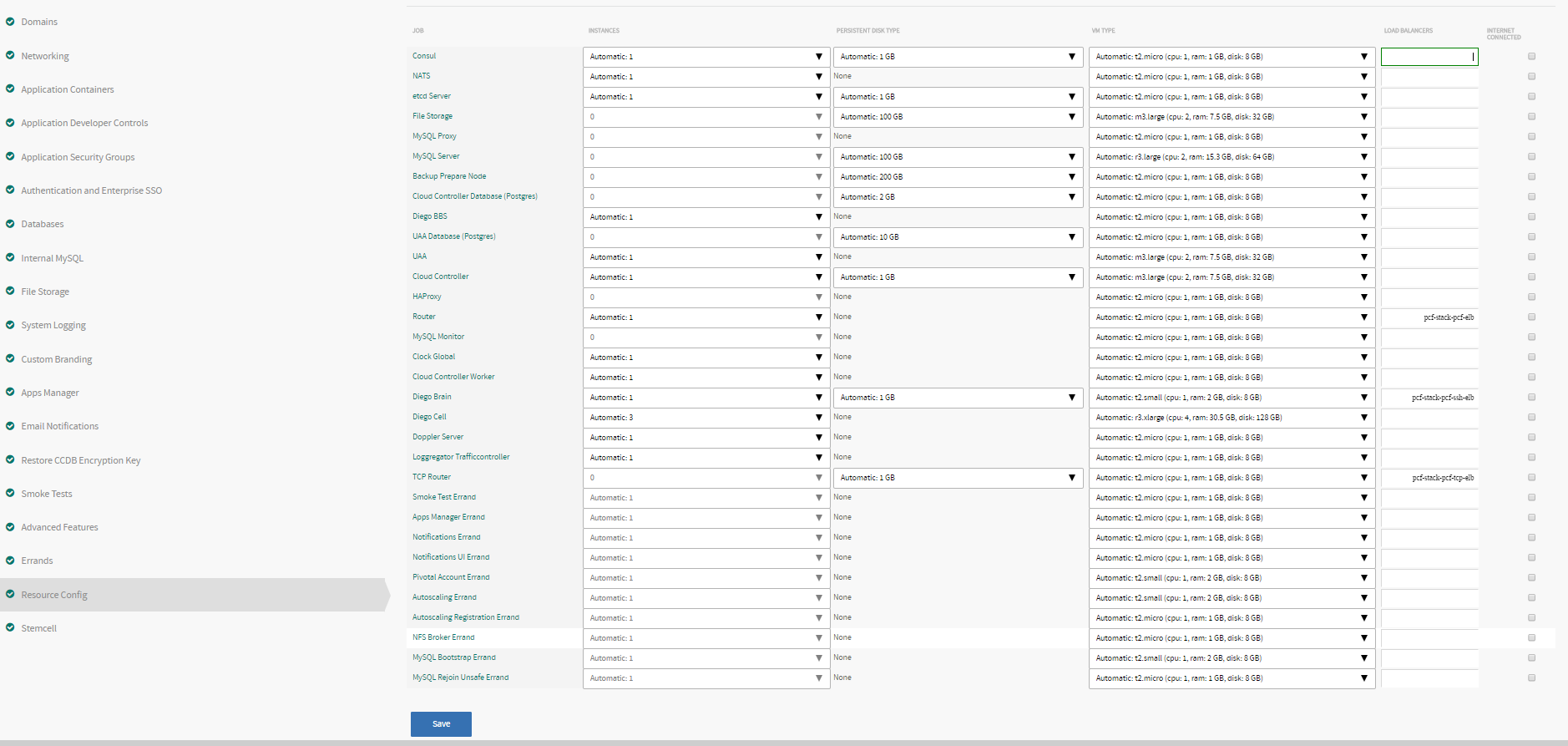


* **Smoke Test Errand** verifies that your deployment can do the following:
  + Push, scale, and delete apps
  + Create and delete orgs and spaces
* **Apps Manager Errand** deploys the Apps Manager, a dashboard for managing apps, services, orgs, users, and spaces. Until you deploy Apps Manager, you must perform these functions through the cf CLI. After Apps Manager has been deployed, Pivotal recommends deselecting the checkbox for this errand on subsequent Elastic Runtime deployments. For more information about the Apps Manager.
* **Notifications Errand** deploys an API for sending email notifications to your PCF platform users.
* **Notifications UI Errand** deploys a dashboard for users to manage notification subscriptions.
* **Pivotal Account Errand** deploys Pivotal Account, a dashboard that allows users to create and manage their accounts. In the Pivotal Account dashboard, users can launch applications, manage their profiles, manage account security, manage notifications, and manage approvals.
* **Autoscaling Errand** enables you to configure your apps to automatically scale in response to changes in their usage load..
* **Autoscaling Registration Errand** makes the Autoscaling service available to your applications. Without this errand, you cannot bind the Autoscaling app to your apps.
* **NFS Broker Errand** enables you to use NFS Volume Services by installing the NFS Broker app in Elastic Runtime.

## **Step 21: Configure Router to Elastic Load Balancer**

1. If you do not know it, find the name of your Elastic Load Balancer (ELB) by clicking **Load Balancers** in the AWS EC2 dashboard. This example shows three ELBs:
   * pcf-stack-pcf-ssh-elb: An SSH load balancer
   * pcf-stack-pcf-elb: A load balancer
   * pcf-stack-pcf-tcp-elb: A TCP load balancer

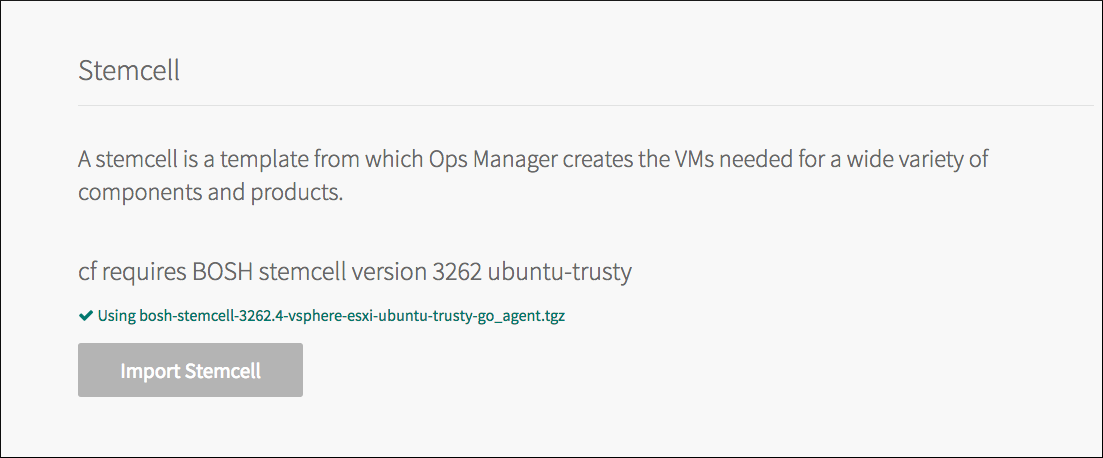


1. In the **Elastic Runtime** tile, click **Resource Config**. 
2. In the **ELB Name** field of the **Diego Brain** row, enter the name of your SSH load balancer. Specify multiple load balancers by entering the names separated by commas. Er aws resource config diego brain
3. In the **ELB Name** field of the **Router** row, enter the name of your load balancer. Specify multiple load balancers by entering the names separated by commas. Er aws resource config router
4. In the **ELB Name** field of the **TCP Router** row, enter the name of your TCP load balancer if you enabled TCP routing in the [Advanced Features](https://docs.pivotal.io/pivotalcf/1-10/customizing/cloudform-er-config.html#advanced-features) pane. Specify multiple load balancers by entering the names separated by commas. Er aws resource config tcprouter
5. Click **Save**.

## **Step 23: Download Stemcell**

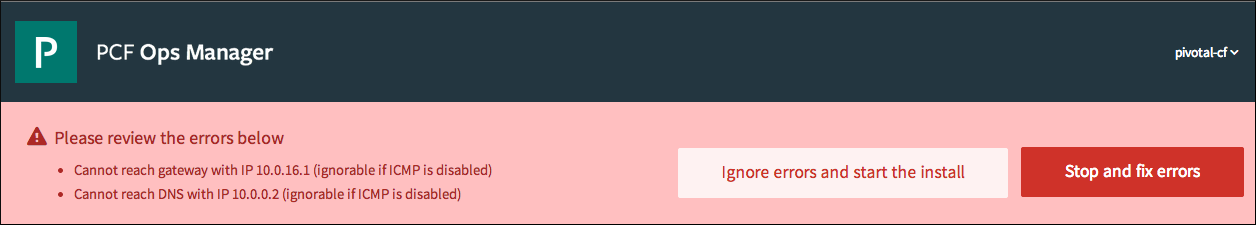
This step is only required if your Ops Manager does not already have the Stemcell version required by Elastic Runtime.

1. Select **Stemcell**.
2. Log into the [Pivotal Network](https://network.pivotal.io/products/pivotal-cf) and click on **Stemcells**.
3. Download the appropriate stemcell version targeted for your IaaS.
4. In Ops Manager, import the downloaded stemcell .tgz file.

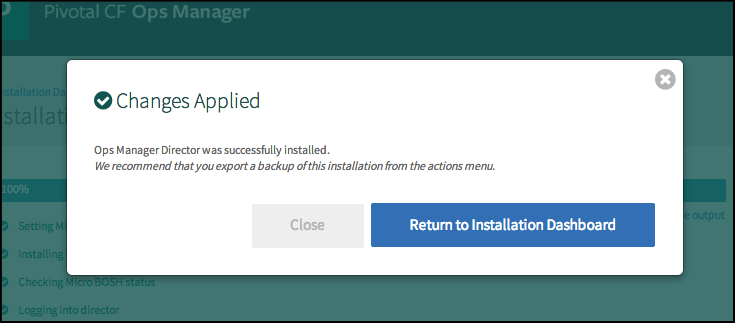


## Step 24: Complete the Elastic Runtime Installation

1. Click the **Installation Dashboard** link to return to the Installation Dashboard.
2. Click **Apply Changes**. If the following ICMP error message appears, click **Ignore errors and start the install**.



The install process generally requires a minimum of 90 minutes to complete. The image shows the Changes Applied window that displays when the installation process successfully completes.



YOU ARE DONE ☺ with Pivotal Elastic Runtime