DXC.technology

DXC Managed Services for AWS Delivery Guide

Version 1.5

April 17, 2018

Table of contents

What's New for DXC Managed Services for AWS Version 1.5

1

DXC Managed Services for AWS Version 1.5 supports the following new features:

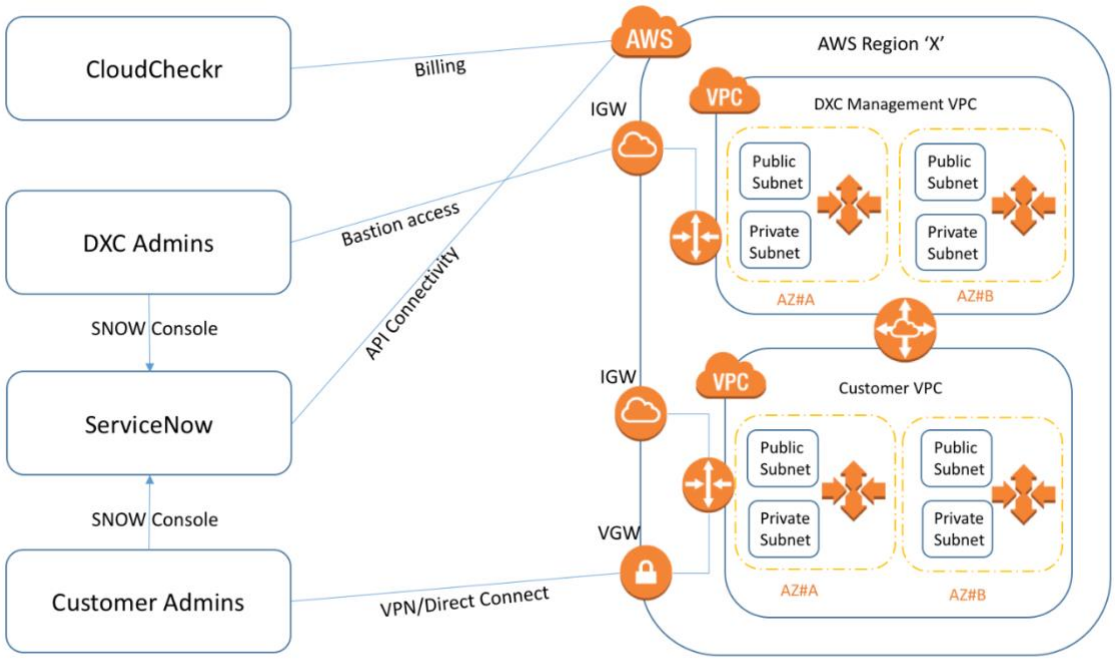
- AWS Config is now enabled on customer AWS accounts in participating regions. Rule violations are streamed to SNS which creates an incident in ServiceNow.
- Support for SUSE 12.

DXC Managed Services for AWS Overview

2

DXC Managed Services for AWS is a feature-rich offering containing Gold Managed Services (billing, logging, end point protection, patching, monitoring, and snapshot of EBS volumes), Silver Plus Managed Services, and automated account configuration with alerting, monitoring, and security policies. The following guide includes the steps to configure a new, empty AWS account with managed services. DXC uses local AWS and other service and SaaS offerings to enable the effective running and maintenance of Gold Managed Services to apply to the client workloads. DXC also stands up a management VPC to enable the hosting of additional management systems (as required on a per client basis).

The diagram below depicts the areas associated with this solution at a high level:



Best Practices

The architecture built by the Managed Services for AWS Quick Start supports AWS best practices for high availability and security:

Multi-Availability Zone architecture intended for high availability

- Isolation of instances between private and public subnets. **Important:** All customer-dedicated public or private subnets must have "*Public*" or "*Private"* in the name string or provisioning will fail from ServiceNow.

Security groups limiting access to only necessary services and ports

Management VPC and secured bastion host instance to facilitate restricted login access for system administrator actions

NAT gateways and proxies to manage Internet access

Standard IAM policies with associated groups and roles that exercise the least privilege

Monitoring, logging, alerts, and notifications for critical events such as logging of root activity, IAM

changes, and changes to logging policies

S3 buckets (with security features enabled) for logging, archive, and application data, including custom lifecycle policies for archiving objects in Amazon Glacier and use of versioning

Prerequisites

For a customer account to be properly configured with managed services, all the requirements described in the prerequisites section must be met.

CloudCheckr Access

The Delivery Engineer must have access to the CloudCheckr account to get the External ID, which is a required parameter of the Master template.

Empty AWS Account

For the automation to properly configure the customer AWS account, the account must be empty or blank.

Access to PROD OBE Account

The Production OBE Account is where all the AMIs and scripts are stored. The Delivery Engineer needs access to this account to start the configuration of the customer account. The account number is 601716130897.

Supported Regions

The following regions are supported for DXC Managed Services for AWS to function properly:

Region Name	Region	AWS Config Notification Support
US East (N. Virginia)	us-east-1	Yes
US East (Ohio)	us-east-2	Yes
US West (N. California)	us-west-1	No
US West (Oregon)	us-west-2	Yes
Canada (Central)	ca-central-1	Yes
Asia Pacific (Mumbai)	ap-south-1	No
Asia Pacific (Singapore)	ap-southeast-1	Yes
Asia Pacific (Sydney)	ap-southeast-2	Yes
Asia Pacific (Tokyo)	ap-northeast-1	Yes

Asia Pacific (Seoul)	ap-northeast-2	No
EU (Frankfurt)	eu-central-1	Yes
EU (Ireland)	eu-west-1	Yes
EU (London)	eu-west-2	Yes

AWS Config evaluations of non-compliances against a set of rules is provided on all supported regions. However, in the regions marked “No”, notification of non-compliances to ServiceNow is not yet supported. As AWS completes the rollout of AWS Step Functions to these regions, AWS Config notification support will be added to them.

South America (São Paulo) sa-east-1 currently does not allow creating NAT Gateways in zone sa-east-1b (the other zones support it) So, our VPC nested template fails to create a NAT Gateway in sa-east-1b zone, thus failing the whole master stack. Currently there is no workaround, therefore this region is not supported.

SSH Keys

As a part of the onboarding process, the Delivery Engineer works with the customer to create or use existing customer-provided SSH keys. These keys are used during the provisioning process to access the Windows or Linux workloads.

To create a key-pair in AWS, see the following guide: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair.

To import a public key into AWS, see the following guide: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#how-to-generate-your-own-key-and-import-it-to-aws.

Security Groups

During the onboarding process, the Delivery Engineer works with the customer to define the required security groups. You must create security groups before an instance is provisioned so that the correct inbound or outbound ports are opened for the Linux or Windows instances. These security groups are applied when an instance is provisioned from the Simple Linux or Simple Windows CloudFormation templates. At a minimum, the inbound port 443 (HTTPS) must be opened on every workload for the managed services to function correctly.

IAM Permissions

To deploy the templates, you must be logged in to the AWS Management Console with Admin IAM permissions for the resources and actions the templates deploy. The *AdministratorAccess* managed policy in IAM provides sufficient permissions, but the organization might choose to use a custom policy with more restrictions.

CSCMS-Delivery-Power-Admin is configured on the Group. There is also a pre-configured Group called *CSCMS-Full-Admin* that has the managed policy *AdministratorAccess* assigned.

Service Limits

For a customer to use their AWS account without service limits, a request must be submitted to AWS to increase the service limits on the account. Failing to do so can result in service limits where resources fail to provision. Submit a case to AWS using the following link to increase service limits: http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html.

Note: Service limits must be increased in each region the customer wants to use.

3 Upgrading from Version 1.4 to Version 1.5

This section describes how to perform an upgrade from Cloud Services for AWS Gold version 1.3 or 1.4 to Gold Cloud Services for AWS versions 1.5.

Important: Gold customers who are using version 1.3 or 1.4 cannot be upgraded to Silver Plus in a later version, and vice versa.

Minimum Version

To perform an upgrade to version 1.5, a customer must have previously deployed either version 1.3 or 1.4. It is not required to upgrade from 1.3 to 1.4 before upgrading to 1.5. A deployment of 1.3 can be upgraded directly to 1.5.

Archiving changes to OBE asset buckets

If you have made any changes to the assets in the OBE buckets that you wish to keep, such as CloudFormation templates, scripts, or Lambda functions, copy the contents of the bucket to another bucket.

To archive changes to OBE asset buckets:

1. Download the following script from the S3 bucket containing the assets you used to launch the master stack for your account, to a system where you have the AWS CLI installed:

```
deploy/utilities/copyBucketContents.sh
```

2. Create a new bucket to hold the contents of the bucket containing the changes you wish to keep.
3. Run the **copyBucketContents.sh** script, supplying the source and destination bucket names, as shown in the following examples:

```
$ copyBucketContents.sh gold.dxc.prod.obe.us-west-1 saved-
gold.dxc.prod.obe.us-west-1
```

OR

```
$ copyBucketContents.sh dxc.customer.config-12345-us-west-1 saved-
dxc.customer.config-12345-us-west-1
```

Deleting the Existing AWS Config setup

The Master Template will create one AWS Config Recorder and one Delivery Channel per region. AWS allows only one AWS Config Recorder and one Delivery Channel per region. If any recorder or channel exists before updating the Master Template, the stack update will fail. To prevent this, follow the steps in the section "*Fixing AWS Config Stack Failure Caused by Existing Recorder and Delivery Channel*".

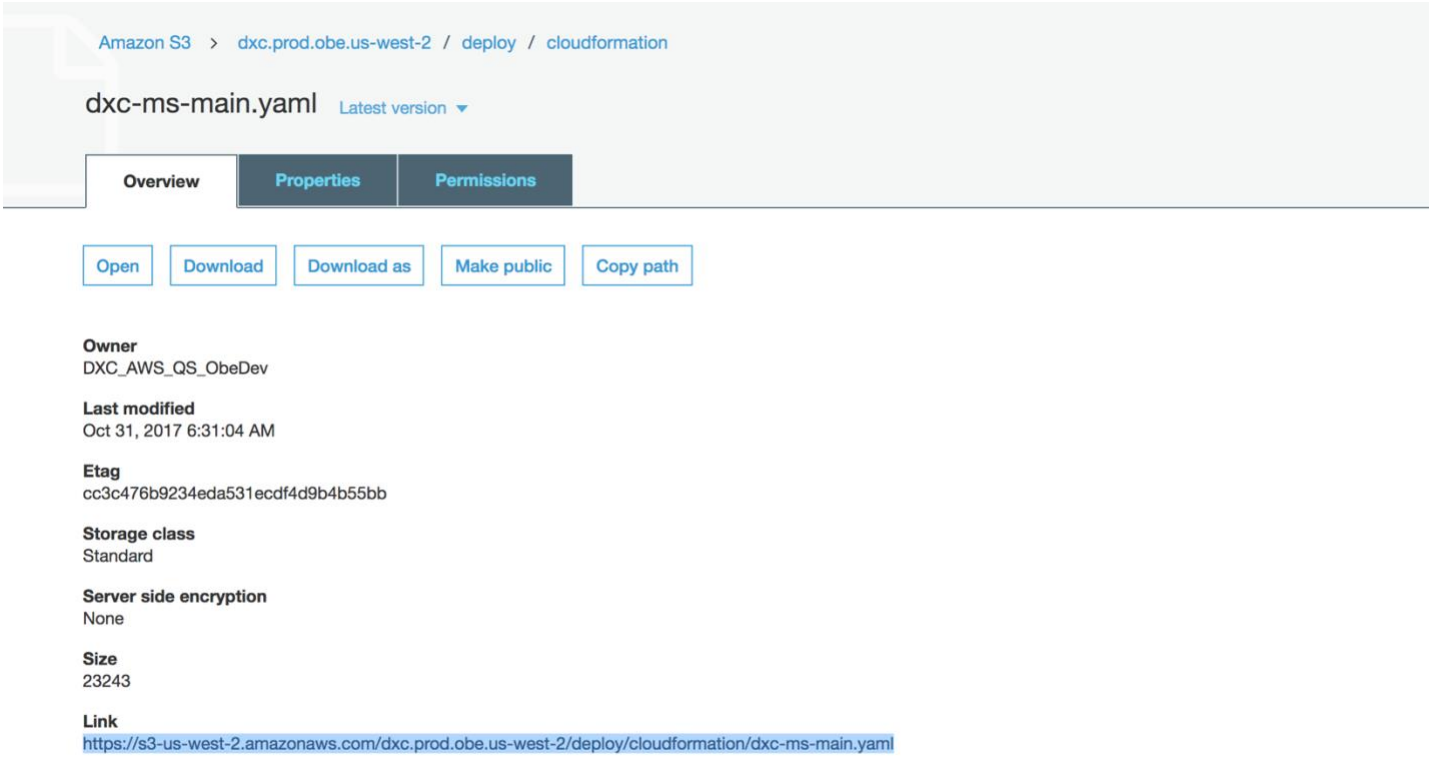
Updating the Master Template

1. Log into your AWS account.
2. Switch over to the region where the master template resides, for example **us-west-2**.
3. Open CloudFormation. You will see a list of stacks that were previously created by the current installation.
4. Open S3 in another browser window. On the **Overview** tab, search for the following bucket: https://s3.console.aws.amazon.com/s3/buckets/gold.dxc.prod.obe/deploy/cloudformation/ This is the bucket where all of the new template files will be pulled from.

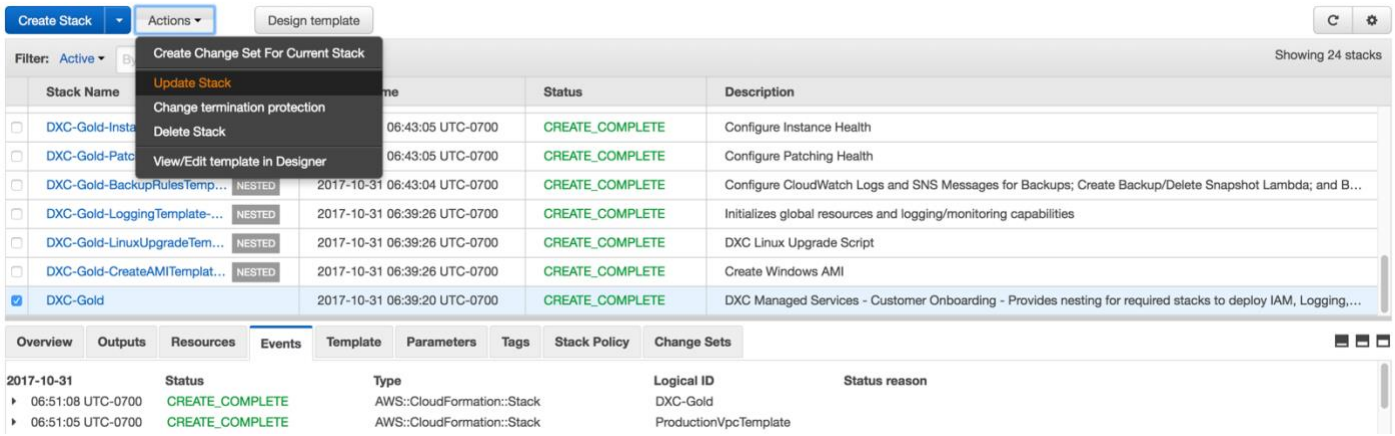
Note: As of the 1.5 release, use one of the following buckets, depending on the customer's current state:

– dxc.prod.obe.REGION_NAME : for release-1.5 (new customers), or customers who were initially deployed at release-1.4 and are now upgrading to release-1.5.
gold.dxc.prod.obe.REGION_NAME : for customers who were initially deployed at release-1.3, have optionally upgraded to release-1.4, and are now upgrading to release-1.5.

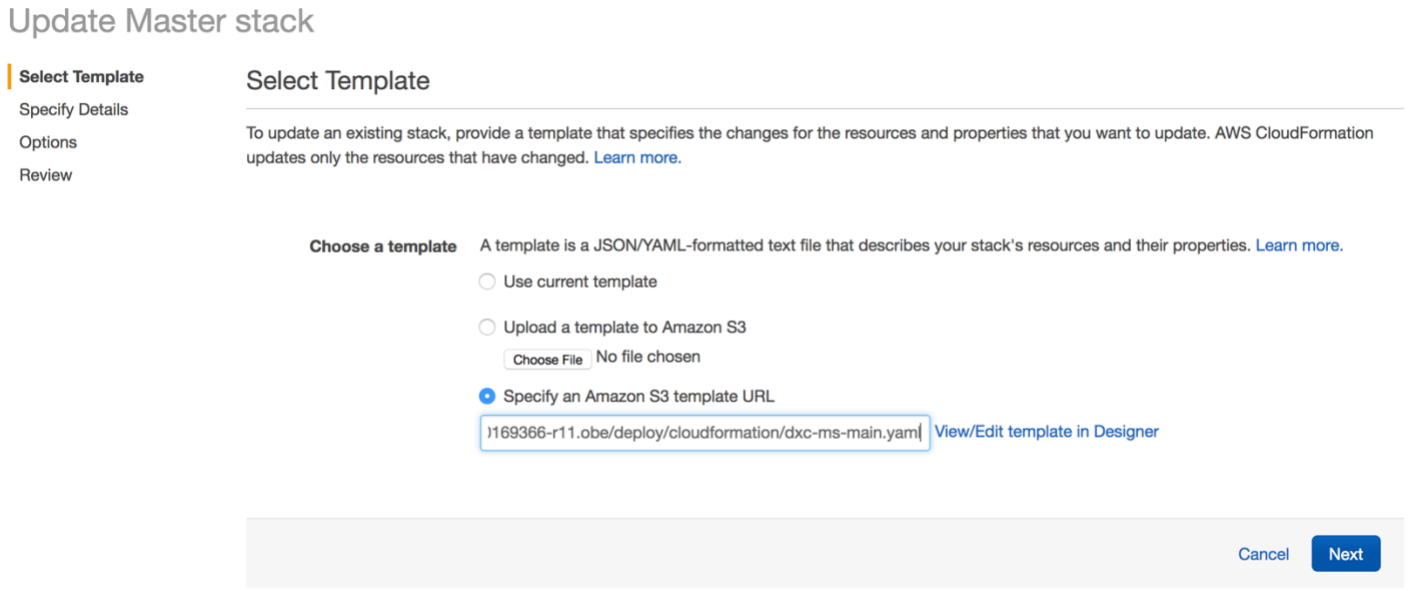
5. Copy the link location for the **Master** template, for example: **dxc-ms-main.yaml**.



6. Go back to CloudFormation.
7. From the list of stacks, select the **Master** stack.
8. From the **Actions** menu, choose **Update Stack**.

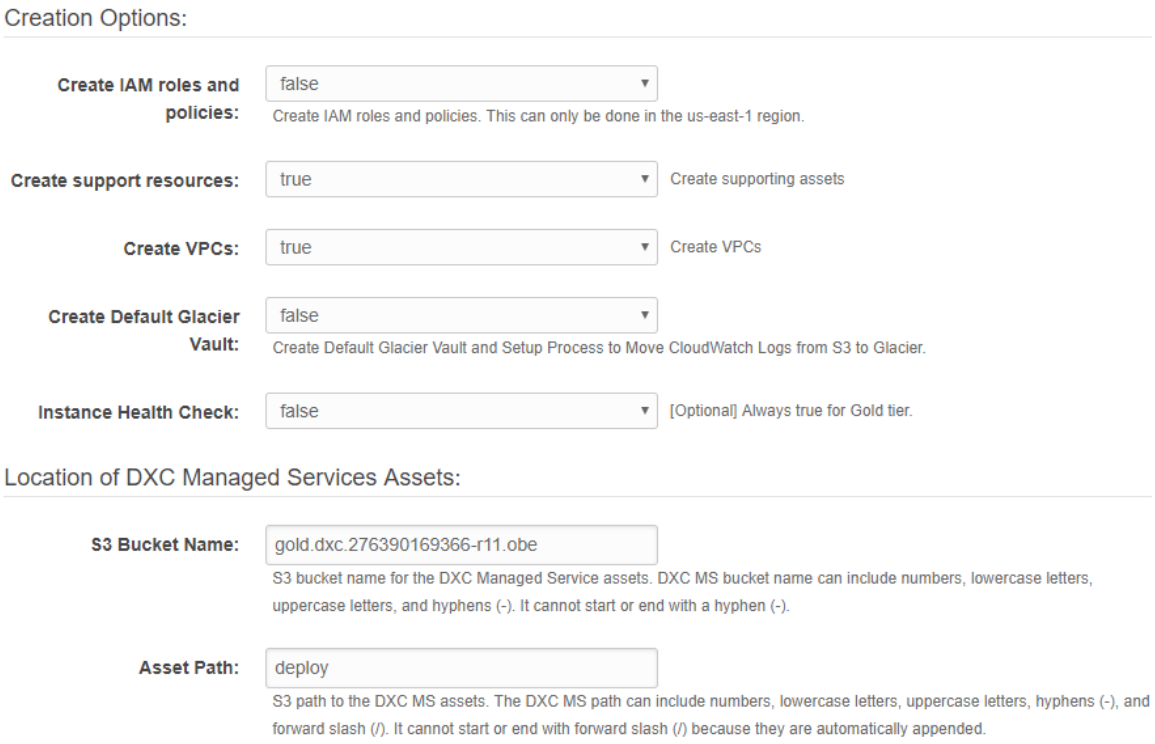


9. Under **Choose a template**, click **Specify an Amazon S3 template URL**, paste the S3 URL for the release-1.5 Master Template, then click **Next**.

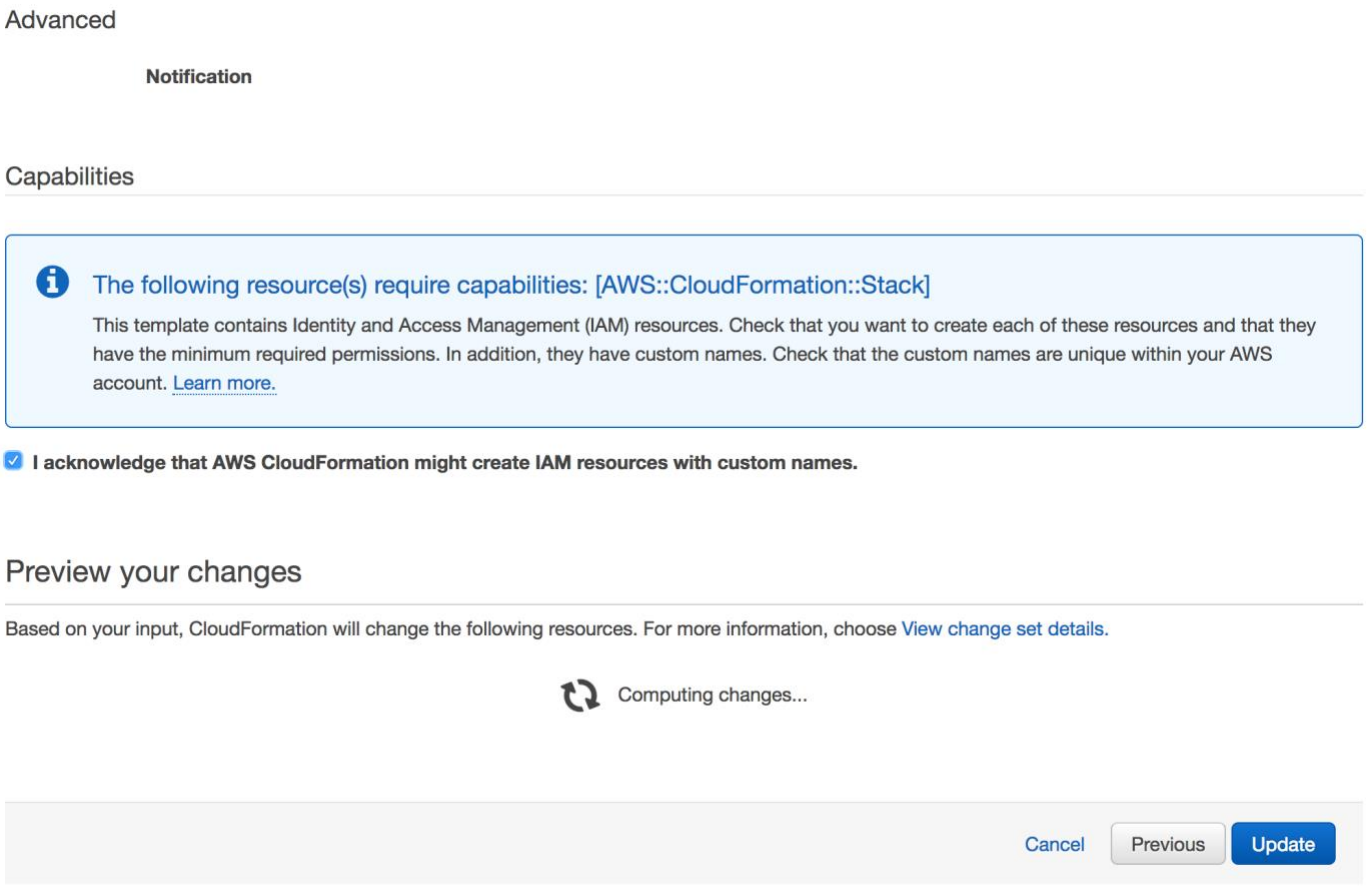


Under **Parameters**, in the **S3 Bucket Name** field, type the name of the S3 bucket for release-1.5, and then click **Next**.
NOTE: The **S3 Bucket Name** and **Asset Path** fields must contain the same values as the original stack or the update will fail.

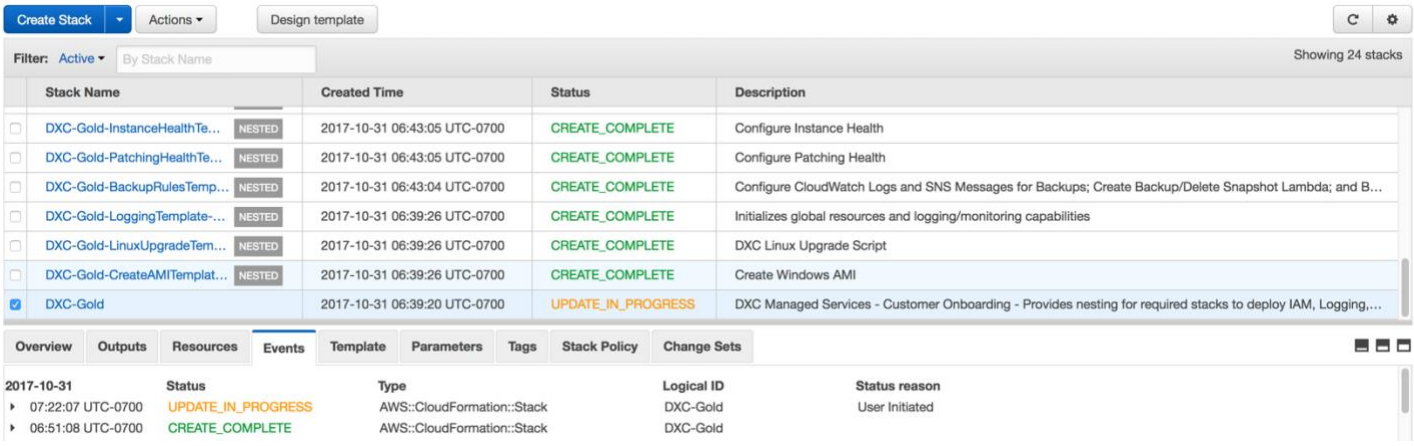
10. On the **Creation Options** page, make sure the correct options are selected, and then click **Next**.
    NOTE: The option **Create IAM roles and policies** should be set to **true** when updating the master stack in **us-east-1**, but it should be set to **false** when updating the stack in all other regions.
    NOTE: Do not change the Service Tier from "Gold" to "SilverPlus" or vice versa, and do not change any of the optional features such as Backups. Such changes are not supported at this time, and deployment errors will result.



11. Enter a patch group name for all **Patch Group** fields, such as the "Patch Group for Windows 2012 R2 Instances" field, even if **Apply Patching** has been set to false. If any of the patch group fields are blank, the deployment will not allow you to proceed. The values you enter do not have to be actual patch group names, but you are encouraged to create a default patch group for every operating system.
12. Click **Next** until you have reached the last page.
13. On the last page, acknowledge that AWS CloudFormation might create IAM resources.



14. Scroll to the bottom of the page, and click **Update**.
    The status of the Master template (DXC-Gold in the example) should display as "UPDATE_IN_PROGRESS".
    Note: In this example, the name of the master template contains "Gold" because it refers to the Gold template, but you can any other name, preferably a name that contains a reference to the offering, such as "Gold," or "Silver Plus."



15. Once the update is completed, all the necessary resources should be updated.
16. Repeat all **Instructions** steps for each region that needs upgrading to the new version.

Updating tags on existing resources during upgrade

During the upgrade to 1.5, certain existing AWS resources created by previous releases will be updated to match how 1.5 resources are managed. Tags applied to EC2 instances, EC2 AMIs, EBS volumes, EBS snapshots, and CloudFormation stacks may be modified. Only resources created by deployments of the Managed Services for AWS product will be modified.

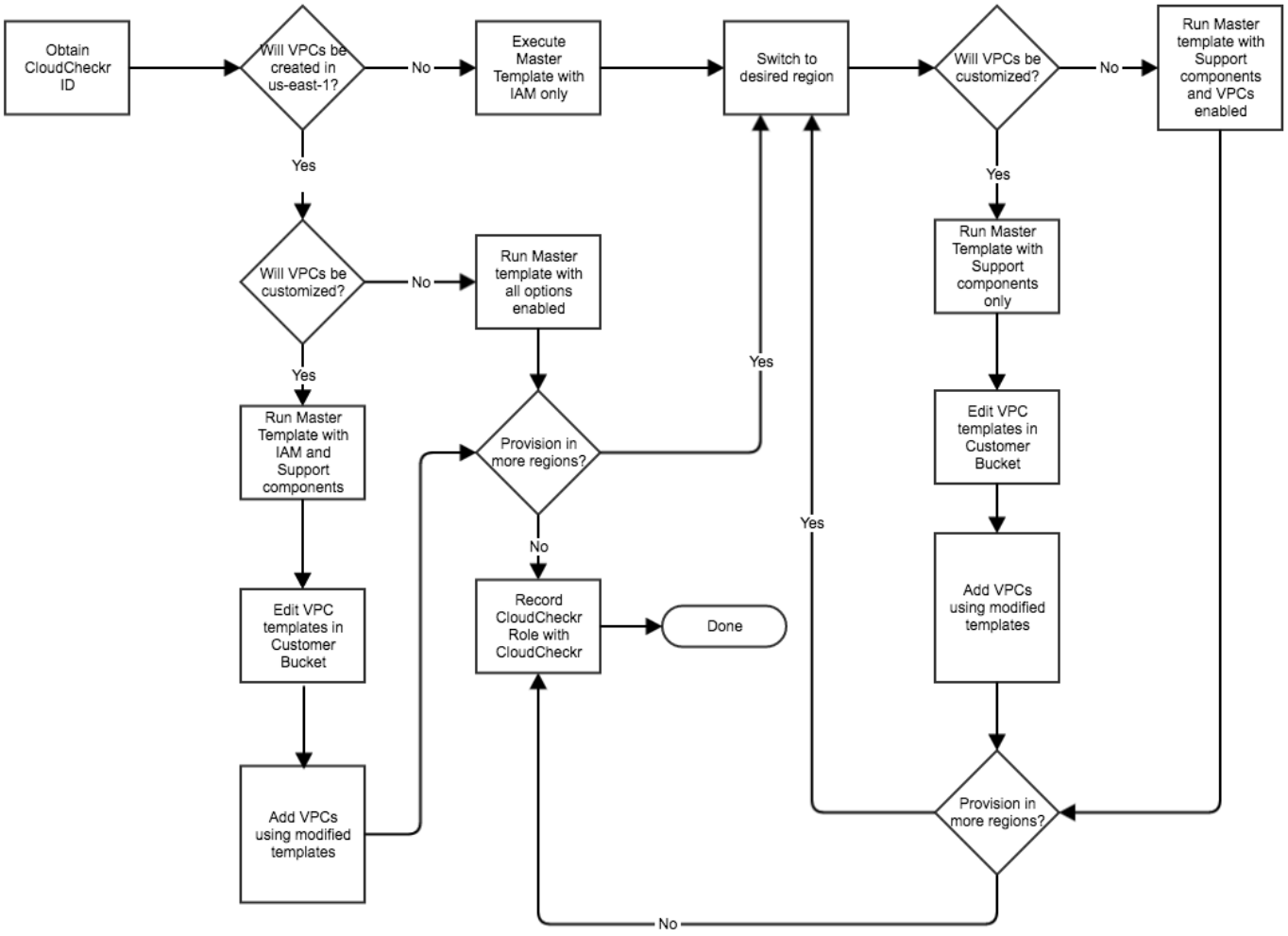
No action is required by service delivery personnel. This is informational only.

4 Configuring a New Customer Account for New Gold and Silver Plus Customers

Configuration of the customer AWS account must be performed on an empty, non-configured AWS account. If the account was previously configured, you must reset the account. The Master template must be run from the supplied S3 link from the Offering Build team. After the Master template finishes executing, all of the scripts are copied to the customer S3 bucket. If the

Delivery Engineer needs to modify the templates, the engineer must download the templates from the customer S3 bucket, modify them, and upload them to the customer S3 bucket using the same naming convention.

Use the following flow chart to determine which path to take to set up the customer account.



Master Template

Use the Master template to install the managed service components into a new customer account. Execute the individual templates in sequence and pass in parameters where needed. The templates currently executed from the Master template are:

IAM - Creates the IAM Roles and Policies used throughout the managed services

Lambda - Creates Lambda functions used internally to support the managed services

Customer Bucket - Creates an S3 bucket in the customer account and copies the DXC Managed Services assets to that bucket

Logging - Creates CloudWatch logging components to monitor resources running in the customer environment

Management VPC - Creates a management VPC

Customer VPC - Creates a VPC where customer workloads will run

- **VPC Peering** - Peering connections are made between the Customer VPC subnets and the Management VPC during creation of the Customer VPC.

Backup - Creates the functions to back up instances

VPC Logs - Creates VPC Flow Logs that are maintained in CloudWatch

Bastion Service - Creates the resources needed to support an on-demand Bastion service

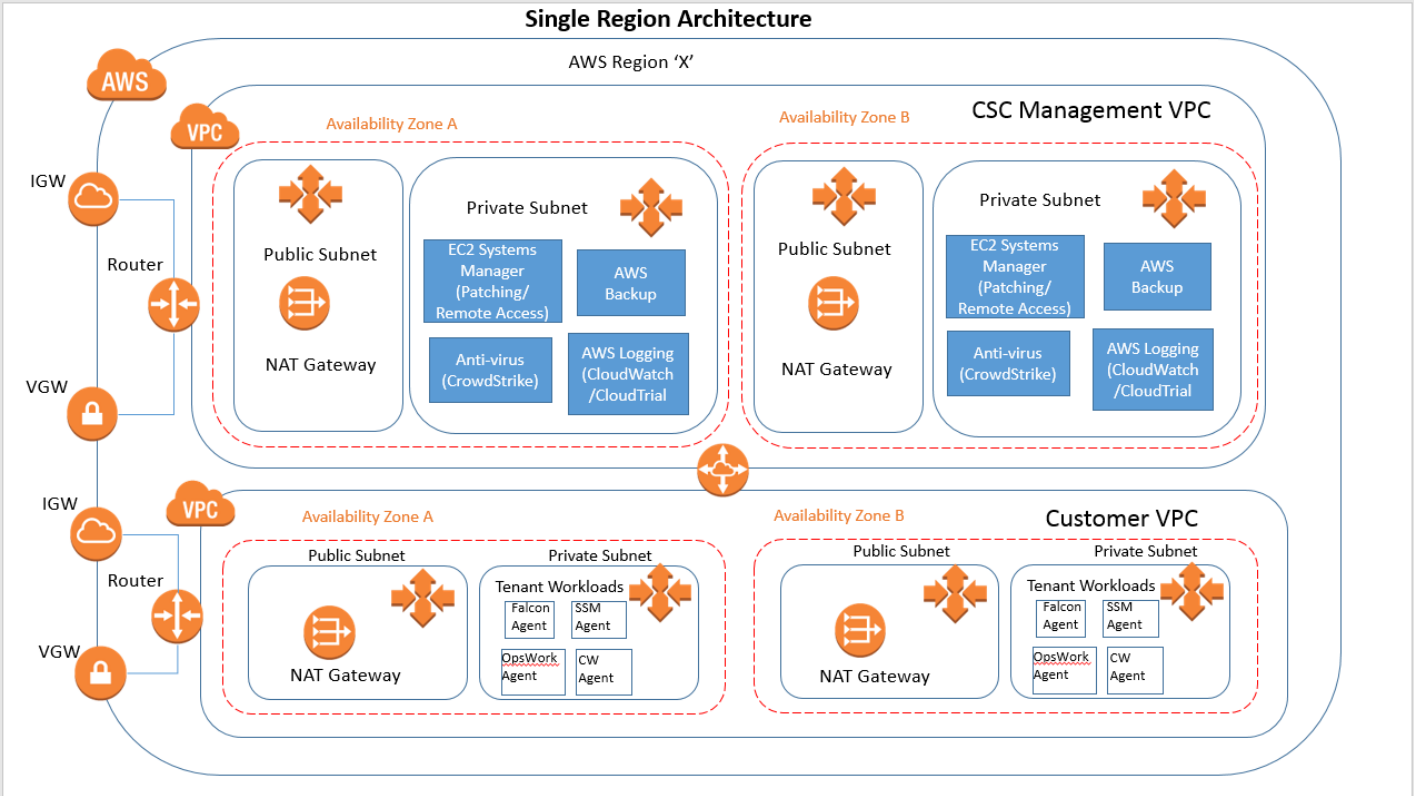
Linux Patching - Creates the resources needed to support patching of Linux instances

Glacier Logs - Creates a Glacier Log for migrated CloudWatch Logs (stored in S3) and configures the process to move these logs from S3 to Glacier as per Retention Intervals settings.

For information on how to use the Master Template to install the managed service components, read *Using the Master Template to Install Managed Service Components into a New Customer Account*. Following the execution of the Master template, several manual processes are required to complete configuration of the managed services. See the *Configuring Gold Managed Services* section for more information.

Logical Configuration

The following diagram shows the configuration of the Management VPC and one Workload VPC in terms of the created AWS resources.



Master Template and Region Requirements

You must first execute the Master template in the **us-east-1** AWS region. The IAM definitions are created in the us-east-1 region and used globally in the account. If the Master template is executed in a different region and has not been executed in the us-east-1 region, some of the subsequent templates will fail. The Master template is designed so that the IAM definitions cannot be added to any region other than us-east-1.

The second portion of the Master template adds the VPCs and other supporting functions, which are region specific and must be executed in each region where the customer deploys instances. When you execute the Master template, adding the VPCs and other functions are optional; therefore, you can add the IAM definitions in us-east-1, but bypass adding the VPCs in that region.

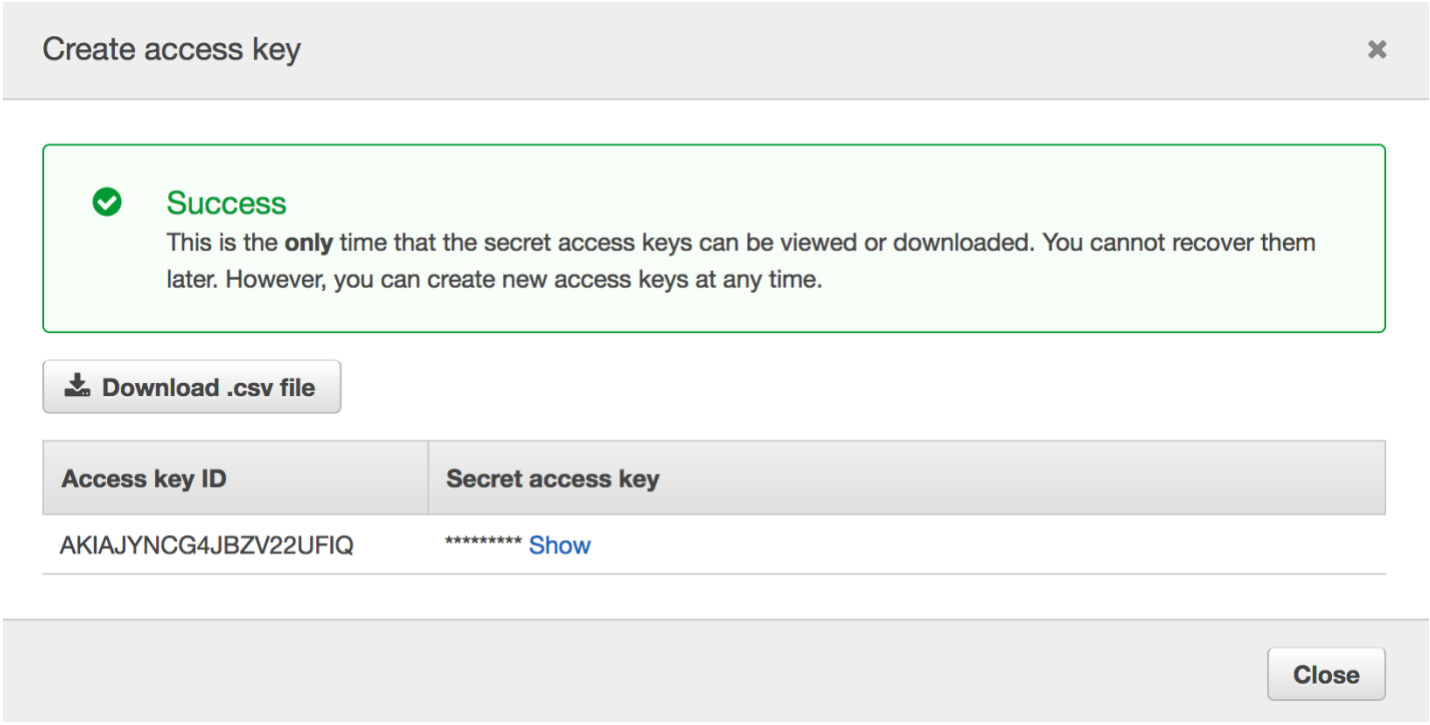
When the configuration of the VPCs must be changed for specific customer requirements, the VPC definitions are edited and the Management VPC template and Workload VPC template are executed manually. This allows the VPC configuration to be modified, tested, deleted, and re-added until the configuration is correct. The procedure for modifying the VPCs is provided below.

IAM Template

The IAM template creates a user account that is used with ServiceNow. A user named SnowUser-<AWS::AccountId> is created. For example, **SnowUser-211682634048** would be the name of the user in the AWS account ID 211682634048. By default, this user account does not have access to the AWS account, but an access key is created for API access to AWS for this user. When the IAM stack is

complete, a set of API keys will need to be generated for the ServiceNow integration. Creation of the API keys is done following this procedure:

1. Log into the AWS Console and switch to the customer account. Select **IAM** from the Services pull-down menu. In the left column, select **Users**. In the User list, select the newly created ServiceNow user. The user name will be *SnowUser-<AWS::AccountId>*.
2. On the Summary page, click on the tab labeled **Security credentials**. In the Access keys section, click on **Create access key**. A set of keys will be generated and the following pop-up will appear:



3. Click on the **Show** link to see the **Secret access key**. You should also download the .csv file to your local computer. It is important to capture the Secret access key at this time as the key will not be available after clicking on the **Close** button.
4. Use the API keys to configure the cloud definition in ServiceNow.

This information is used to configure the cloud account in ServiceNow and access AWS. Currently, this user account has ***full access to all functions in AWS*** so these keys should be handled properly. Best practices suggest that the keys should be rotated at least once per year: a new set should be generated and configured into ServiceNow and the old keys removed from AWS.

Supported Features for Gold and Silver Plus Managed Services

DXC Cloud Services for AWS supports Gold and Silver Plus Managed Services.

The following table shows a matrix of the supported features.

Feature	Gold	Silver Plus
Backup	Supported by default	Can be enabled
Patching	Supported by default	Can be enabled
Instance Health Monitoring	Supported by default	Can be enabled
Endpoint Protection	Supported by default	Can be enabled

Note: For Silver Plus accounts, all features listed in the above table can be enabled during workload creation. Backup, patching and instance health monitoring can also be enabled when the initial master template is executed. For more information, read "*Using the Master Template to Install Managed Service Components into a New Customer Account*."

Using the Master Template to Install Managed Service Components into a New Customer Account

Use the Master template to install the managed service components into a new customer account. Execute the individual templates in sequence and pass in parameters where needed. The templates currently executed from the Master template are:

Template	Description
IAM	Creates the IAM Roles and Policies that are used throughout the managed services.
RHEL AMI Automation	Configures the automation document for creating a RHEL AMI.
Windows AMI Automation	Configures the automation document for creating a Windows AMI.
Silver Plus Service Test	[Optional] Only run this template if Silver Plus tier is selected.
Glacier Logs	Configures a Glacier vault when required.
Lambda	Creates Lambda functions used internally to support the managed services.
Customer Bucket	Creates an S3 bucket in the customer account and copies the DXC Managed Services assets to that bucket.

Template	Description
Logging	Creates CloudWatch logging components to monitor resources running in the customer environment.
Management VPC	Creates a management VPC.
Customer VPC	Creates a VPC where customer workloads will run.
Backup	Creates the functions to back up instances.
VPC Logs	Creates VPC Flow Logs that are maintained in CloudWatch.
Bastion Service	Creates the resources needed to support an on-demand Bastion service.
Linux Patching	Creates the resources needed to support the patching of Linux instances.

Following the execution of the Master template, several manual processes are required to complete the configuration of the managed services.

Deleting the existing AWS Config setup

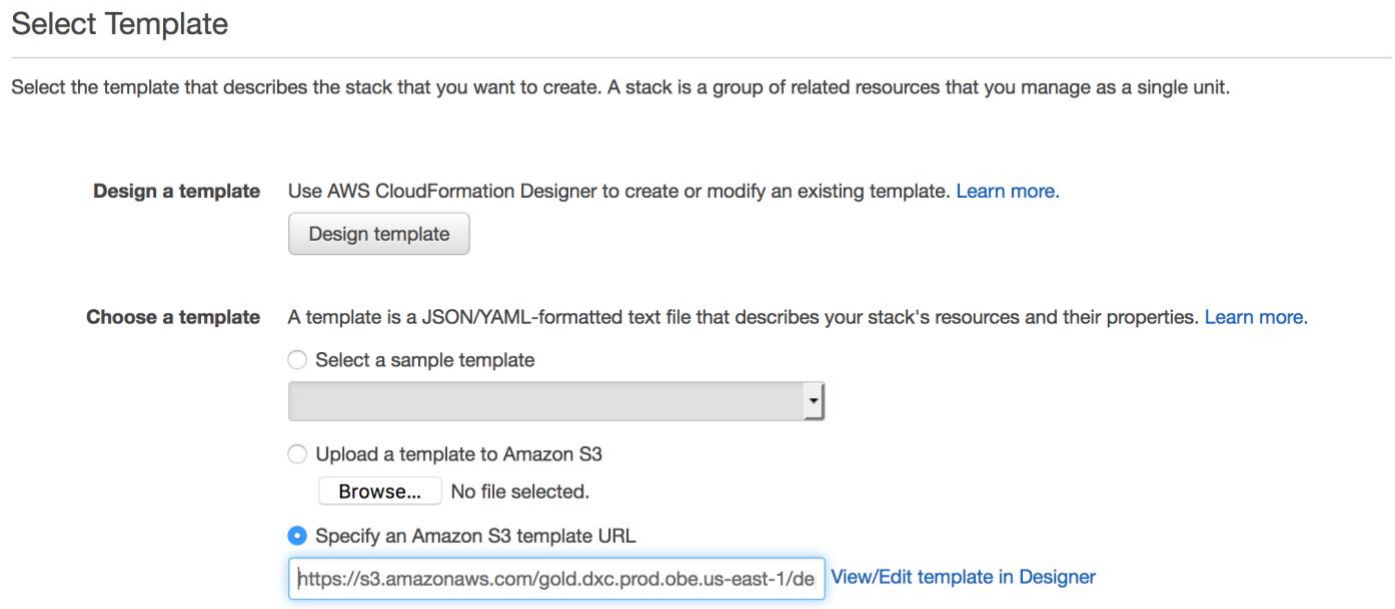
The Master Template will create one AWS Config Recorder and one Delivery Channel per region. AWS allows only one AWS Config Recorder and one Delivery Channel per region. If any recorder or channel exists before running the Master Template, the stack creation will fail. To prevent this, follow the steps in the *Fixing AWS Config stack failure caused by existing Recorder and Delivery* Channel section.

Running the Master Template

This example describes how to run the Master template in **the us-east-1** region, add only the IAM roles and policies, and then run it again in the **us-west-2** region to create the VPC and other components.

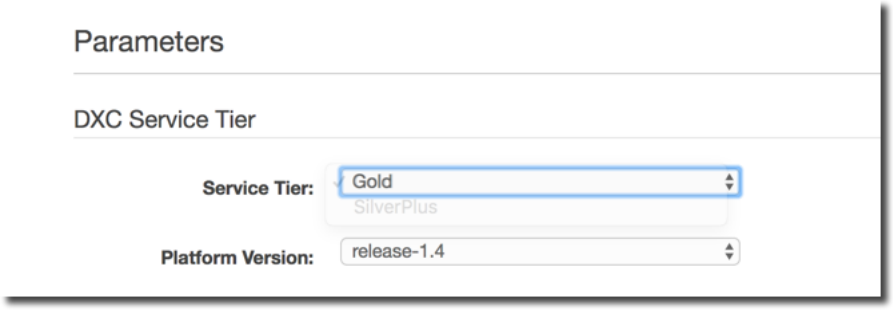
Note: Most templates and URLs contain "Gold" in their names, but this doesn't mean that a URL or template is limited to Gold customers only, as these items apply to Silver Plus customers as well.

1. Log into the AWS console.
2. Select the CloudFormation AWS service and verify you are in the **us-east-1** AWS region.
3. Click **Create Stack**.
4. On the **Select Template** screen, click **Specify an Amazon S3 template URL**.

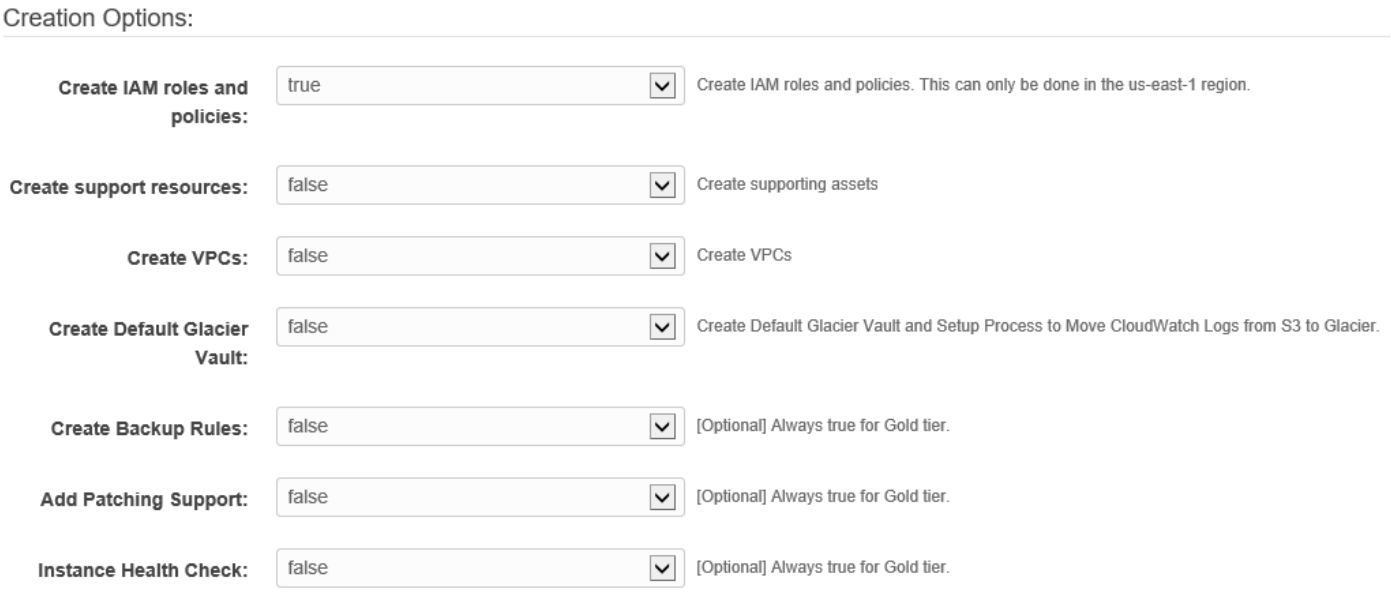


5. Enter the template URL, for example: https://s3.amazonaws.com/dxc.prod.obe.us-east-1/deploy/cloudformation/dxc-ms-main.yaml
6. Click **Next**.
7. On the **Specify Details** page, enter **Master** for the **Stack name**.

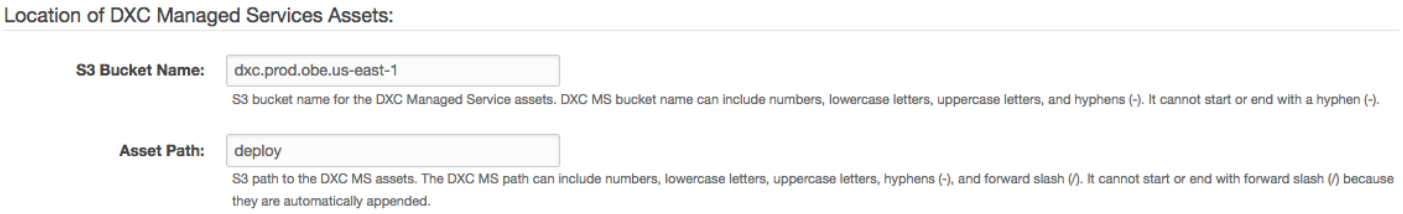
8. Select the service tier you want to use for the **Service Tier** option. Select the **Platform Version** which indicates the release version of the service. Currently there is only one choice.



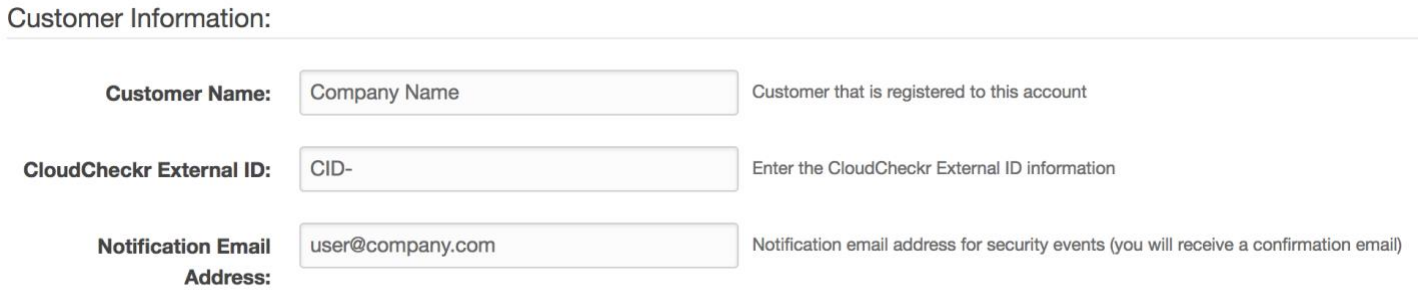
9. To create the global IAM resources, set the **Create IAM roles and policies** option to ***true*** in the **Creation Options** section. Because you are only adding the IAM roles and policies, set the other Creation Options to ***false***. Some options may only exist for Silver Plus.



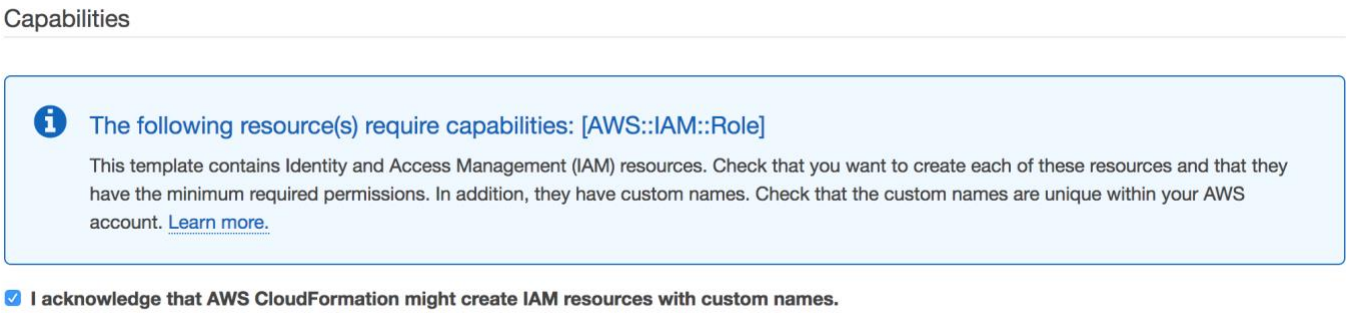
10. Under **Location of DXC Managed Services Assets**, verify that the **S3 Bucket Name** is **dxc.prod.obe.us-east-1** and that **Asset Path** is set to **deploy**.



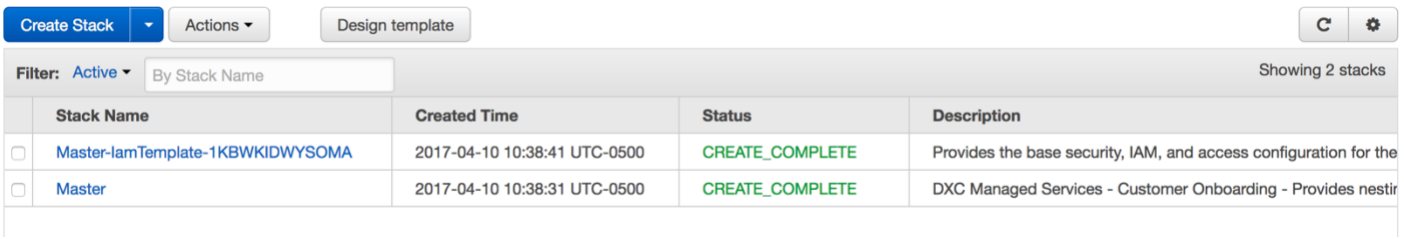
11. Scroll down to the **Customer Information** section.



12. Enter the **Customer Name**, **CloudCheckr External ID**, so that the IAM roles and policies are created.
    In this case, the **Notification Email Address** is not set up so it does not need to be filled in at this point.
13. Scroll down the page and click **Next**.
    Note: The VPC parameters are not used at this point.
14. On the **Options** page, click **Next**.
    You do not need to add anything on this page.
15. On the **Review** page, look at the values that were entered for the parameters.
16. Scroll to the bottom of the **Review** page and select **I acknowledge that AWS CloudFormation might create IAM resources with custom names**.



17. Click **Create**.
    The Master template starts execution. When it completes, you should see stacks similar to the following except that the names will be slightly different:



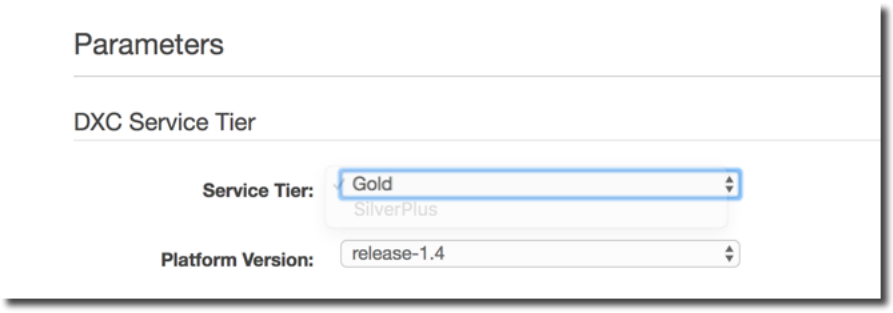
18. Switch to the region where you will deploy workloads and set up the VPC.
    In the example, **us-west-2** was used.
19. Switch to the **us-west-2** region and select the CloudFormation service.
20. Click **Create Stack**.
21. Click **Specify an Amazon S3 template URL**.
22. Enter the following URL:

https://s3.amazonaws.com/dxc.prod.obe.us-west-2/deploy/cloudformation/dxc-ms-main.yaml

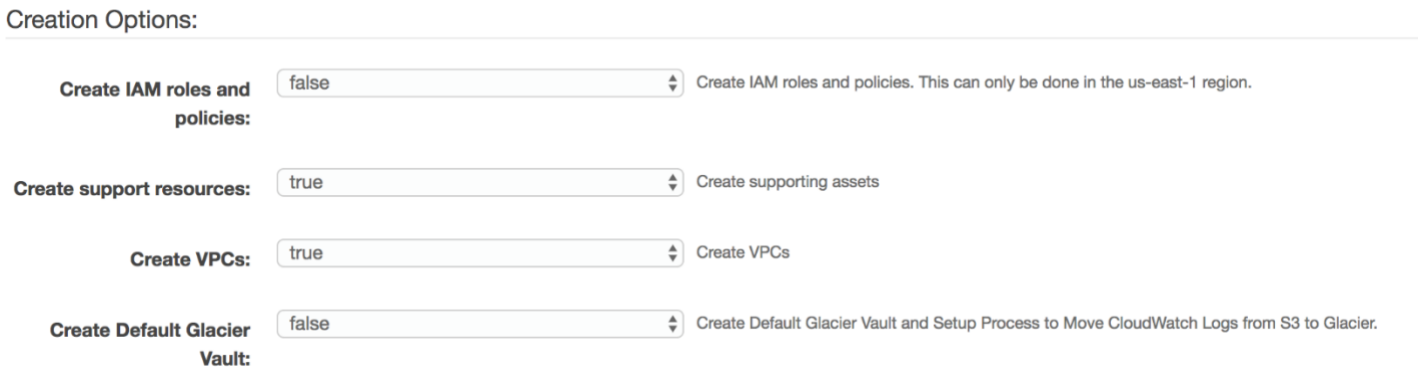
The name of the bucket varies depending on which region you are running the Master Template in. The regions and associated buckets currently supported are:

- dxc.prod.obe.us-east-1
- dxc.prod.obe.us-east-2
- dxc.prod.obe.us-west-1
- dxc.prod.obe.us-west-2
- dxc.prod.obe.ca-central-1
- dxc.prod.obe.ap-south-1
- dxc.prod.obe.ap-southeast-1
- dxc.prod.obe.ap-southeast-2
- dxc.prod.obe.ap-northeast-1
- dxc.prod.obe.ap-northeast-2
- dxc.prod.obe.eu-central-1
- dxc.prod.obe.eu-west-1
- dxc.prod.obe.eu-west-2

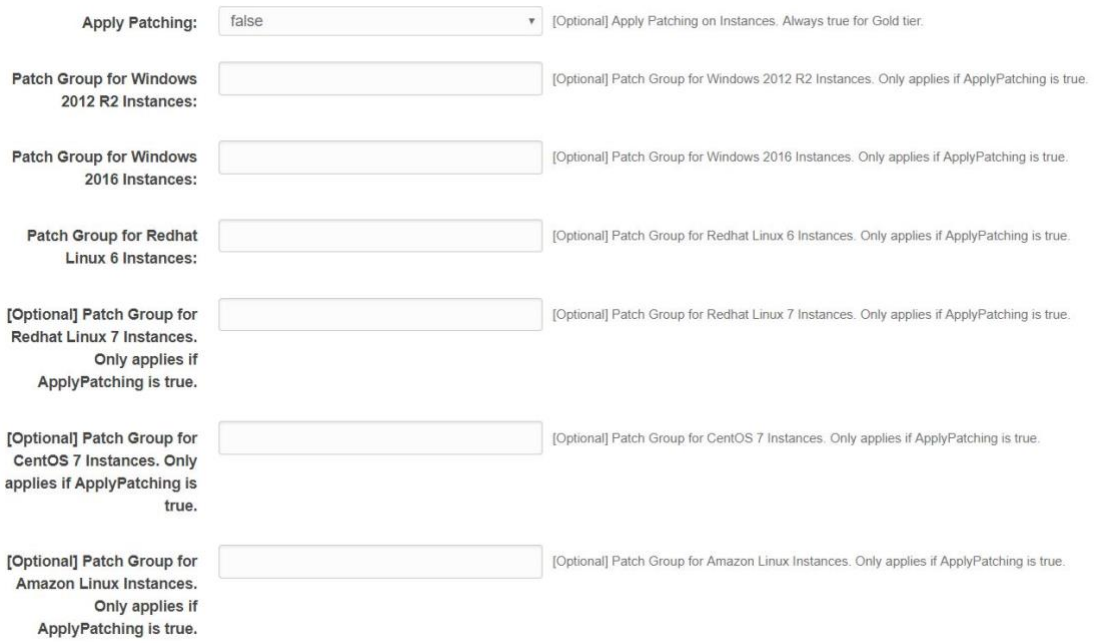
23. Click **Next**.
24. Select the same service tier you selected for us-east-1 for the **Service Tier** option. Select the same **Platform Version** as us-east-1 as well.



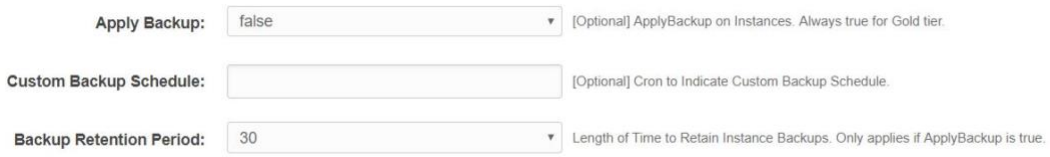
25. In the **Creation Options** section shown below configure the following items:
    - Set **Create IAM roles and policies** option to ***false***.
    - Set **Create support resources** and **Create VPCs** options to ***true***.
    - If you want to configure Glacier to move CloudWatch logs from S3, set the **Create Default Glacier Vault** option to ***true***.
    - The **Instance Health Check** option is optional. Set it to true if needed.



26. **[Silver Plus Only]** Select your choice of Patching and Patch Group options from the list below, **Apply Patching** is true by default for the Gold tier.

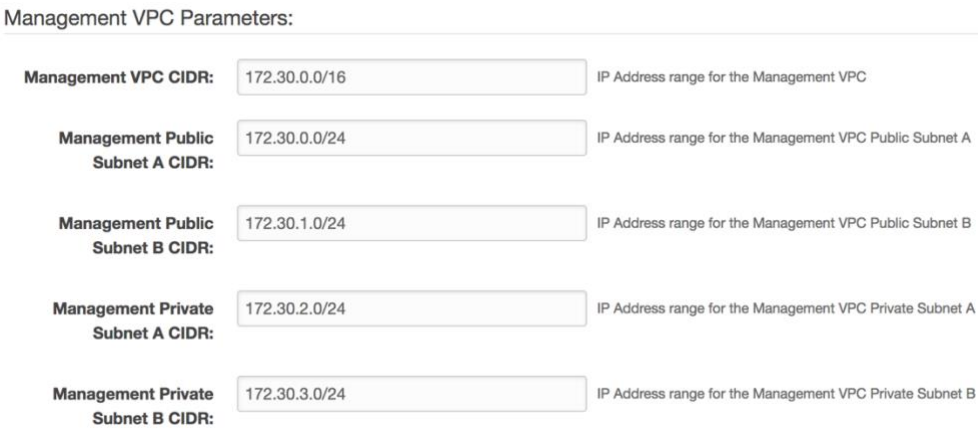


27. **[Silver Plus Only]** Select your choice of **Apply Backup**, **Backup Retention Period and Custom Backup Schedule** options. This option is true by default for the Gold tier. Note that **Custom Backup Schedule** and **Backup Retention Period** works for both Gold and SilverPlus services.

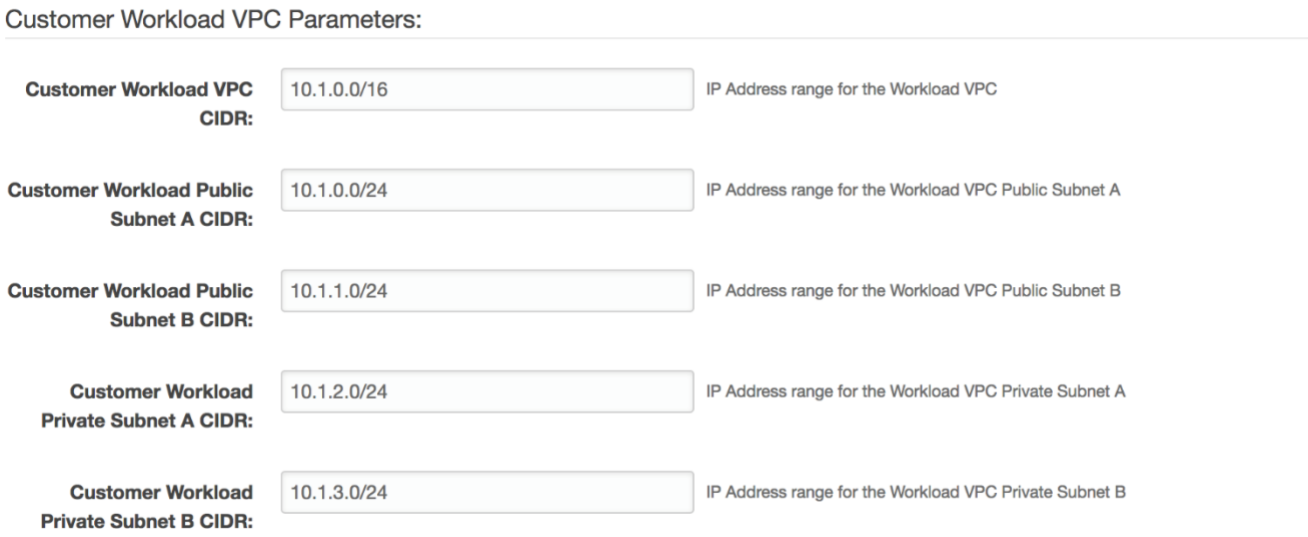


28. [Silver Plus Only] Select your choice of **Apply EndPoint Protection** and **Apply Monitoring Options**. This option is true by default for the Gold tier.

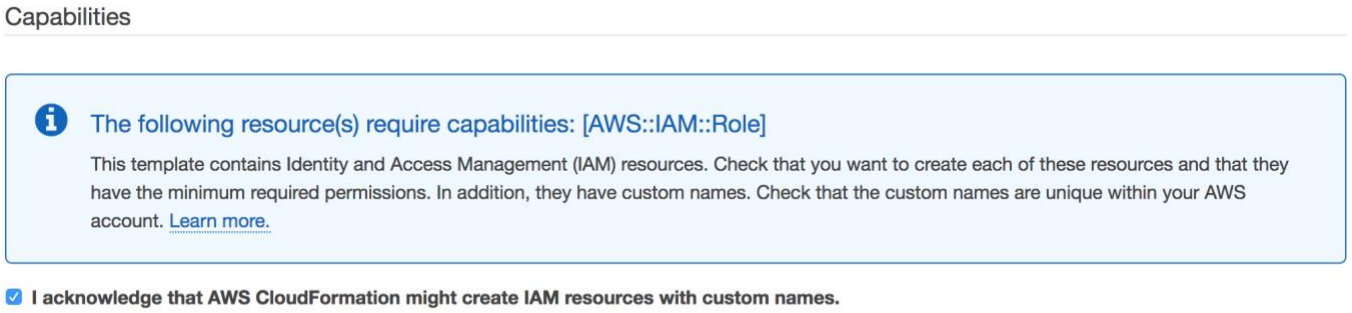
29. Scroll down to the **Management VPC Parameters** section. Enter the desired CIDR block for the VPC and subnet address ranges.



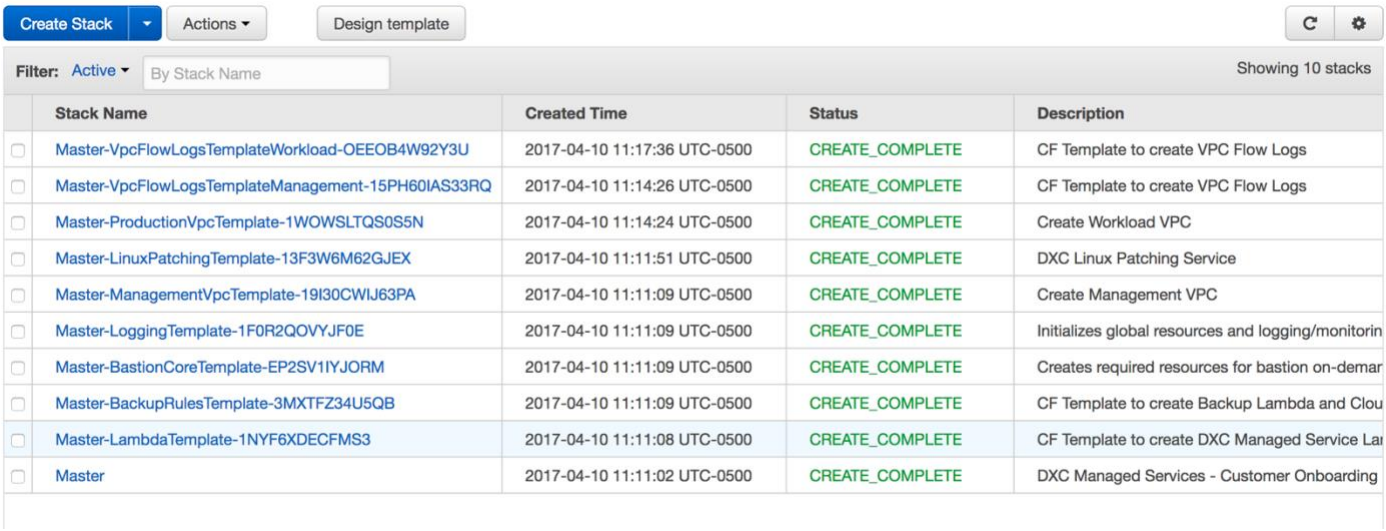
30. Scroll down to the **Workload VPC Parameters** section. Enter the desired CIDR block for the VPC and subnet address ranges.



31. Click **Next**.
32. On the **Options** page, click **Next** again.
33. On the **Review** page, look at the values entered for the parameters.
34. Scroll to the bottom of the page and click **I acknowledge that AWS CloudFormation might create IAM resources with custom names**.

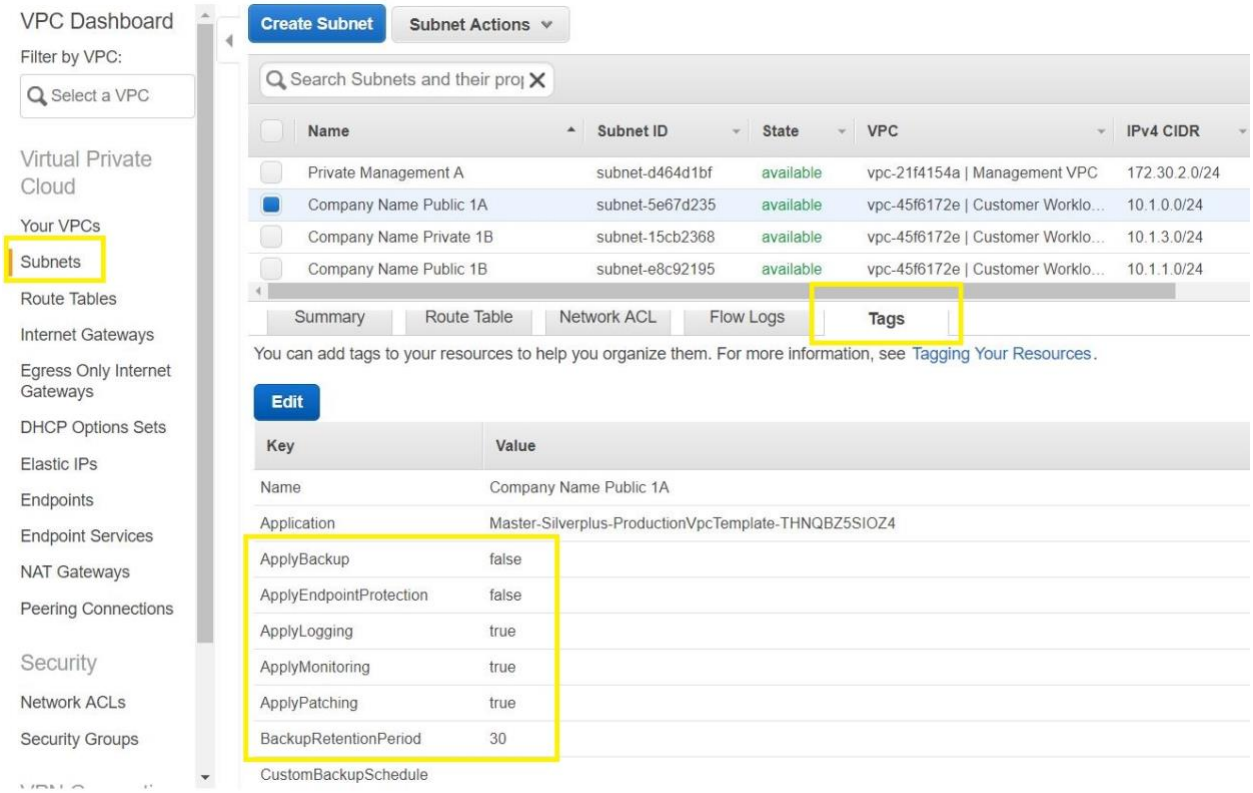


35. Click **Create**.
    The Master template starts execution. When it completes, you should see stacks similar to the following with different names:



Stack Name	Created Time	Status	Description
Master-VpcFlowLogsTemplateWorkload-OEEOB4W92Y3U	2017-04-10 11:17:36 UTC-0500	CREATE_COMPLETE	CF Template to create VPC Flow Logs
Master-VpcFlowLogsTemplateManagement-15PH60IAS33RQ	2017-04-10 11:14:26 UTC-0500	CREATE_COMPLETE	CF Template to create VPC Flow Logs
Master-ProductionVpcTemplate-1WOWSLTQS0S5N	2017-04-10 11:14:24 UTC-0500	CREATE_COMPLETE	Create Workload VPC
Master-LinuxPatchingTemplate-13F3W6M62GJEX	2017-04-10 11:11:51 UTC-0500	CREATE_COMPLETE	DXC Linux Patching Service
Master-ManagementVpcTemplate-19I30CWIJ63PA	2017-04-10 11:11:09 UTC-0500	CREATE_COMPLETE	Create Management VPC
Master-LoggingTemplate-1F0R2QOVYJF0E	2017-04-10 11:11:09 UTC-0500	CREATE_COMPLETE	Initializes global resources and logging/monitorin
Master-BastionCoreTemplate-EP2SV1IYJORM	2017-04-10 11:11:09 UTC-0500	CREATE_COMPLETE	Creates required resources for bastion on-demar
Master-BackupRulesTemplate-3MXTFZ34U5QB	2017-04-10 11:11:09 UTC-0500	CREATE_COMPLETE	CF Template to create Backup Lambda and Clou
Master-LambdaTemplate-1NYF6XDECFMS3	2017-04-10 11:11:08 UTC-0500	CREATE_COMPLETE	CF Template to create DXC Managed Service La
Master	2017-04-10 11:11:02 UTC-0500	CREATE_COMPLETE	DXC Managed Services - Customer Onboarding

The account is now configured. Patching, Logging, Backup and Monitoring Tags are applied at the subnet level. Any instance(s) launched under these subnets (Private and Public) will inherit these settings automatically. There is no need to define these options again while launching simple workload templates.



To launch a workload, you must do the following:

- Create one or more key-pairs. A key-pair is needed during provisioning and when you connect to the instance using SSH or RDP.
- Create one or more security groups. The security groups are specified on each instance when provisioned.

Modifying the VPCs

In some cases, you might need to change the VPC configuration from the model shown above. Especially, you might need to modify the Workload VPC to accommodate customer requirements. This section describes changes that might be necessary and how to make these modifications.

Management VPC

The Management VPC is configured with two public subnets and two private subnets with one public subnet and one private subnet in different Availability Zones. The public subnets are connected to the Internet through an Internet Gateway resource. The private subnets are also connected to the Internet using a NAT gateway. Peering is configured by default between the Management VPC and the Workload VPC.

Currently the Management VPC is used only to support the Bastion Server that is used to connect to customer workloads when you troubleshoot issues with instances. In the future, additional management software might be configured to run in the Management VPC. For now, you might want to remove some of the resources created in the Management VPC to save costs. The following list shows some of the resources that can be removed. You can install the Management VPC as configured above and manually remove the resources that are not needed.

1. **Remove private subnets.** Currently these subnets do not support additional management services so they can be removed. In addition, the NAT Gateways supporting a connection to the Internet can be removed. If one or more of the private subnets are removed, you must first remove the VPC peering configured with these subnets.
2. **Remove a public subnet.** Currently there are two public subnets. To reduce costs, you can remove one of these subnets.
3. **Change the Availability Zones where the subnets are located.** Currently, the first two Availability Zones are used to hold the subnets.

You might also want to add additional subnets to the Management VPC. To do this, you must determine the proper CIDR for the subnet and add routes to the proper route tables.

Workload VPC

You probably need to customize the Workload VPC for each customer. This VPC is currently configured identically to the Management VPC; that is, two public subnets and two private subnets spread across two Availability Zones. Common changes you can make to the Workload VPC are similar to the Management VPC:

1. Removing or adding private subnets
2. Removing or adding public subnets
3. Changing the Availability Zones used for the subnets
4. Modifying the peering

The CIDR blocks used for each subnet are parameters to the template that creates the Workload VPC and its subnets. If you do not need to change the number of subnets, no modifications are needed. You need to specify only the desired CIDR blocks during VPC creation.

Making Changes

The first way to make changes to the VPC configuration is to add the template as it currently exists and manually modify the resources as desired. Perform this task from the AWS Console. See the AWS documentation if you decide to manually modify the configuration. While this probably provides an expedient way to make changes to the VPC configurations, it is not recommended. Instead, you should modify the VPC templates and create the VPC configuration using the modified CloudFormation templates. This way the desired configuration is documented by the templates.

To modify the CloudFormation templates, follow this procedure. This procedure assumes that you have an empty account where no DXC Gold Managed Services were previously installed. If you already executed some of these steps, skip to the appropriate starting step.

Creating the IAM Roles and Policies

To create IAM roles and policies:

1. Locate the Master Template in the us-east-1 region. The S3 bucket name should be **gold.dxc.prod.obe.us-east-1**. The file is in the **/deploy/cloudformation** directory and is named **dxc-ms-main.yaml**.
2. Execute this template in the us-east-1 region with the following parameters:
   - Create IAM roles and policies: true
   - Create support resources: false
   - Create VPCs: false
   - Gold S3 Bucket Name: gold.dxc.prod.obe.us-east-1
   - Gold Asset Path: deploy
   - Customer Name: Enter the customer name

- Switch to the region where workloads will be launched if this is different from us-east-1.

Creating the Supporting Resources

To create the supporting resources:

3. Locate the Master template again. The S3 bucket name will be **gold.dxc.proj.obe.<region>**. For example, in the us-west-2 region, the bucket name would be: **gold.dxc.prod.obe.us-west-2**.
4. Execute the Master template in the desired region. For example, in the us-west-2 region you would specify the parameters as:
   - Create IAM roles and policies: false
   - Create support resources: true
   - Create VPCs: false
   - Gold S3 Bucket Name: gold.dxc.prod.obe.us-west-2 (or the desired region)
   - Gold Asset Path: deploy
   - Customer Name: Enter the customer name
   - Notification Email Address: Enter the email address where CloudWatch alarms will be sent.
5. The above process creates a Customer Bucket in the same region with a copy of all Gold Managed Services assets; in particular, the CloudFormation templates used to create the VPC. The customer bucket will be named: *dxc.customer.config-<AWS::AccountId>-<AWS::Region>*. For example: **dxc.customer.config-211682634048-us-west-2**.

Editing the VPC Templates

Edit the Management VPC, the Workload VPC templates, or both as necessary in the Customer S3 bucket. You must download the template to a local computer, edit it, and reload it into the Customer S3 bucket.

The Workload VPC template contains comments to show where modifications might commonly be made. For example, to change the Availability Zone of a subnet, the current template contains this content:

Workload VPC Template – AZ Selection

```
# Public subnet A:
   PublicSubnetA:
     Type: AWS::EC2::Subnet
     DependsOn:
     - VPC
     Properties:
       VpcId:
         Ref: VPC
       CidrBlock:
         Ref: PublicSubnetCIDRA
# 1 >>>
       AvailabilityZone:
         Fn::Select:
         -  '0'
         - Fn::GetAZs:  ''
# 1 <<<
```

This template section shows the beginning definition of the first public subnet in the Workload VPC. The Availability Zone selected is simply the first Availability Zone in the region. A customer might want to select a different Availability Zone, possibly using the direct name of the Availability Zone. For example, in the us-west-2 region, the options would be us-west-2a, us-west-2b, and us-west-2c.

A different section of the template contains the definition of the second private subnet:

Workload VPC – Second Subnet

```
# Private subnet B:
   PrivateSubnetB:
     Type: AWS::EC2::Subnet
     DependsOn:
     - VPC
     Properties:
       VpcId:
         Ref: VPC
# 1 >>>
       AvailabilityZone:
         Fn::Select:
         -  '1'
         - Fn::GetAZs:  ''
# 1 <<<
       CidrBlock:
         Ref: PrivateSubnetCIDRB
       Tags:
       - Key: Application
         Value:
           Ref: AWS::StackName
       - Key: Network
         Value: Private Subnet B
       - Key: Name
         Value:
           Ref: PrivateSubnetNameB
       - Key: SubnetType
         Value: Private
```

Possible modifications in this section include:

1. Elimination of the second private subnet.
2. Duplication of the subnet to create a third private subnet.
3. Customization of the Availability Zone (as in the previous example).

The template also contains the basic routing definitions needed to support connections to the Internet. You might need to change these definitions to match the new subnet configuration.

Whatever changes are required, you can modify the Management VPC template and the Workload VPC template in the customer's S3 bucket and create the desired resources from the modified templates.

Adding the Management VPC

Because the configuration of the Management VPC will be different from what the Master template provides for parameters, you must add the Management VPC template individually rather than using the Master template.

1. Locate the Management VPC CloudFormation template in the Customer's S3 bucket and enter the parameters based on the modified configuration. At a minimum, specify a CIDR for the VPC and at least one public subnet.
2. Click through the remaining screens and build the Management VPC stack.

3. Verify that the desired Management VPC configuration is present in AWS. If it is not, you can delete the stack for the Management VPC and edit and add the template again.

Adding the First Workload VPC

Because the configuration of the Workload VPC will be different from what the Master template provides for parameters, you must add the Workload VPC template individually rather than using the Master template.

1. Locate the Workload VPC CloudFormation template in the customer's S3 bucket and enter the parameters based on the modified configuration. At a minimum, specify a CIDR for the VPC and at least one private subnet.
2. Click through the remaining screens and build the Workload VPC stack.
3. Verify that the desired Workload VPC configuration is present in AWS. If it is not, you can delete the stack for the Workload VPC and edit and add the template again.
4. Repeat the Workload VPC process for additional Workload VPCs that you need to create. You might need to duplicate the Workload VPC template and maintain a second Workload VPC template with different configuration options than the first Workload VPC.

VPC Flow Logs

If the VPCs are added manually using the procedure above (where the Master template is not used), you must also manually configure the VPC logging. The CloudFormation template used to create the flow logs is named **vpc-flow-logs.json** and can be found in the customer's S3 bucket with the VPC creation templates. The VPC Flow Logs template takes three parameters:

- **VPC ID** - The ID of the VPC where flow logs will be created.
- **Traffic Type** - The type of traffic to log. This parameter defaults to ALL. Other options are ACCEPT and REJECT.
- **Flow Log Name** - The name of the CloudWatch Log Group where the logs will be stored. This parameter defaults to **VPCFlowLogs**. Continue to use this Log Group name or you can use a different Log Group name for each VPC if desired.

Execute the VPC Flow Logs template for each VPC created, which includes the Management VPC and all Workload VPCs.

By default, log data is stored in CloudWatch indefinitely. You can configure how long to store log data in a log group. Any data older than the current retention setting is automatically deleted. You can change the log retention for each log group at any time.

To change the logs retention setting:

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, click **Logs**.
3. For **Expire Events After**, choose the retention setting to change.
4. In the Edit Retention dialog, select a log retention value.
5. Choose **OK**.

VPC Peering

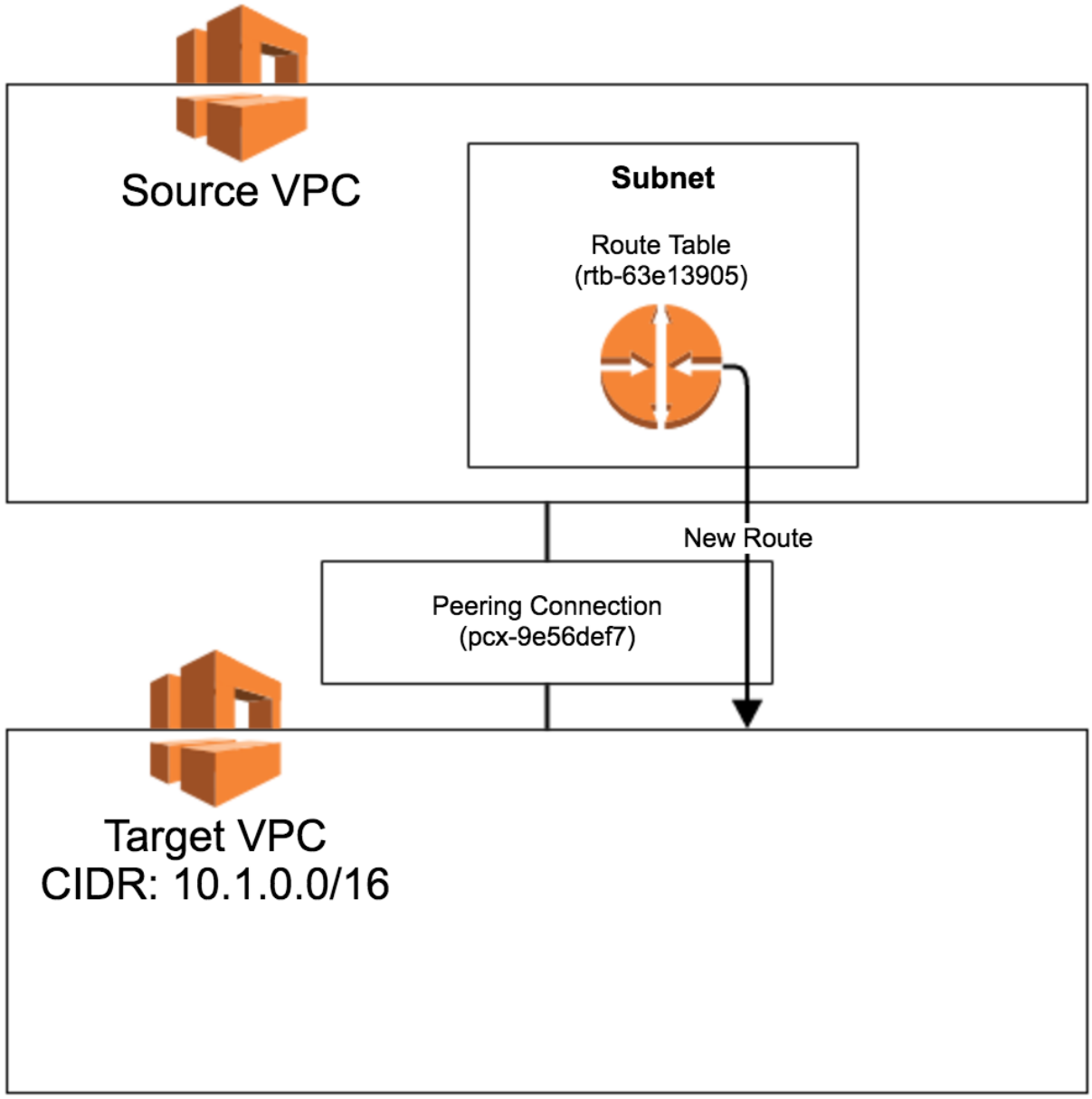
Using the default setup in the Gold Managed Services CloudFormation templates, peering is configured between the Management VPC and the Workload VPC according to the "Logical Configuration" section above. This results in six routes: three from the Management VPC subnets and three from the Workload VPC subnets:

1. Management Public Router to Workload VPC
2. Management Private Router A to Workload VPC
3. Management Private Router B to Workload VPC
4. Workload Public Router to Management VPC
5. Workload Private Router A to Management VPC
6. Workload Private Router B to Management VPC

A simple CloudFormation template sets up these routes and is called **PeeringRoute.yaml**. This template takes three parameters:

1. **Peering Connection ID** - The ID of the existing peering connection created between the VPCs.
2. **Target CIDR** - The CIDR of the VPC where the traffic will be targeted.
3. **Source VPC Route Table ID** - The ID of the subnet's route table where the route will be added.

Following is a simplified diagram of the structure:



The Peering Connection must already exist. Example parameters might look like this:

- **Peering Connection:** pcx-9e56def7
- **Route Table:** rtb-63e13905
- **CIDR:** 10.1.0.0/16

The Peering Connection is normally created by the Workload VPC template. The peering routes are normally created by the same Workload VPC template. If you change the Management VPC or Workload VPC, you might need to modify the peering. For each subnet, you must determine if a peering route

should be constructed to the other VPC. Using the parameters of the target VPC, the source route table, and the peering connection, create the new route.

Fixing AWS Config Stack Failure Caused by Existing Recorder and Delivery Channel

AWS allows only one AWS Config Recorder and one Delivery Channel per region. This document provides the steps to fix the issue caused by the failed execution of a CFT that enables the AWS Config service for a customer (for a specific region).

Scenario

AWS Config stack fails due to the following:

An AWS Config Recorder already exists for a customer in a given region.

A Delivery Channel already exists for a customer in a given region.

The above scenario can occur if:

The AWS Config service was enabled manually through the AWS console. Note that turning Recorder '*Off*' does not delete the existing recorder from the region for a particular customer.

AWS Config service was enabled via executing the main stack (dxc-ms-main.yaml), but the stack fails after it enabled and configured AWS Config for a customer for a particular region. In this case the main stack should be rolled back, but for some odd reason rollback does not delete all the resources completely.

AWS Recorder and Delivery Channel cannot be deleted using the AWS console. The AWS CLI must be used to delete the Recorder and Delivery Channel. Please use the following steps to do so.

Deleting the AWS Config Recorder

To delete an AWS Config Recorder:

1. Issue the following AWS CLI command to view the existing Config Recorder:

```
aws configservice describe-configuration-recorders --profile <profilename>
```

2. Copy the name of the Recorder from the output provided by the above command as shown in the screenshot below:

3. Issue the following command to delete the Recorder, using the name listed above (in step 2):

```
aws configservice delete-configuration-recorder --configuration-recorder-name
default --profile <profilename>
```

4. The above command deletes the existing Config Recorder. Run the **describe-configuration-recorders** command to verify. You should see no Config Recorder for the customer for the region as shown in the screenshot below.

```
$ aws configservice describe-configuration-recorders --profile devseoul
{
    "ConfigurationRecorders": []
}
```

Deleting the Delivery Channel

To delete a Delivery Channel:

1. Issue the following AWS CLI command to view the existing Delivery Channel:

```
aws configservice describe-delivery-channels --profile <profilename>
```

2. Copy the name of the Delivery Channel from the output provided by the above command as shown in the screenshot below:

```
$ aws configservice describe-delivery-channels --profile devseoul
{
    "DeliveryChannels": [
        {
            "snsTopicARN": "arn:aws:sns:ap-northeast-2:225992052696:config-topic",
            "name": "default",
            "s3BucketName": "master-gold-tsyed-seoul-rawsconfigbucket-1it7pcsxtq2eo"
        }
    ]
}
```

3. Issue the following command to delete the Delivery Channel, using the name listed above (in step 2):

```
aws configservice delete-delivery-channel --delivery-channel-name default --
profile <profilename>
```

4. The above command deletes the existing Delivery Channel. Run the **describe-delivery-channels** command to verify. You should see no Delivery Channel for the customer for the region as shown in the screenshot below.

```
$ aws configservice describe-delivery-channels --profile devseoul
{
    "DeliveryChannels": []
}
```

Try running the AWS Stack (or the main stack that runs the AWS Stack as well) again

Configuring Billing

To learn about how to configure billing, read the *CloudCheckr AWS New Client Onboarding Guide*.

5

Creating S3 Buckets for the Customer Account

This section describes the S3 buckets that are created in the customer's account.

6

Customer Bucket

The Customer bucket is created during onboarding and contains a full copy of the DXC Gold Managed Service AWS components. These components include the CloudFormation templates used to create the infrastructure in the customer's account. These templates are provided so that they can be modified. The VPC template usually requires modification to support the customer's desired configuration.

A Customer bucket is created in each region where instances will be provisioned. The name of the bucket follows this convention:

dxc.customer.config-<AWS::AccountId>-<AWS::Region>

For example:

dxc.customer.config-211682634048-us-west-2

Archive Logs S3 Bucket

The Archive Logs bucket is used to hold archived logs. The default configuration is to hold archived logs for 90 days and then move the archived logs to Glacier.

CloudTrail S3 Bucket

The CloudTrail bucket holds CloudTrail logs. The logs are archived to the Archive Logs S3 bucket and then eventually to Glacier according to the retention settings.

Creating IAM Roles and Policies

7

This section describes the IAM Policies, Roles, Profiles and Users, and User Groups created by the IAM template. If the Master template has run successfully, this section can be skipped. This section describes what the IAM template configures and how it can be run individually outside of the Master template.

Permissions

A Delivery Engineer needs Admin privileges to the customer AWS account to run the template individually.

Requirements

There are no CloudFormation template prerequisites for the IAM template. You must create a CloudCheckr ID for the customer before you create the IAM stack.

Creation

The IAM stack is normally created as part of the Master template execution. You can also create the IAM stack by executing the *IAM.yaml* CloudFormation template. There is a single parameter:

CloudCheckr Customer ID - This is an ID created through the CloudCheckr web portal. A unique ID is created for each customer account.

The IAM stack is always created in the **us-east-1** region. This is done to facilitate a lookup process to retrieve the roles during the creation of other Gold Managed Services.

CloudCheckr Integration

When the IAM stack is complete, a CloudCheckr role is created. This role has read access to most services in AWS. The role ARN is generated as part of the IAM stack:

The ARN for the CloudCheckr role is fed back to the CloudCheckr web portal to register the role for this account with CloudCheckr.

Other Roles

The following roles are created by the IAM template:

- **Default Instance Role** - This role is assigned to instances (through the Default Instance Profile) to allow instances access to some of the AWS functions.
- **Linux Patching Role** - This role allows for patching of Linux instances.
- **Windows Patching Role** - This role allows for patching of Windows instances.
- **VPC Flow Logs Role** - This role allows creation of VPC Flow Logs.

User Groups

The following User Groups are created by the IAM template:

- **Delivery Power Admin** - Provides the following privileges:
  - arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
  - arn:aws:iam::aws:policy/AmazonEC2ReadOnlyAccess
  - arn:aws:iam::aws:policy/AWSCloudTrailFullAccess
  - arn:aws:iam::aws:policy/AWSOpsWorksFullAccess
- **Full Admin** - Provides the arn:aws:iam::aws:policy/AdministratorAccess privilege
- **Delivery Read Only**
- **Customer Read Only**

Policies

The following Policies are created by the IAM template:

- **Linux Patching Policy** - Assigned to the Instance Role
- **Windows Patching Policy** - Assigned to the Instance Role
- **CloudWatch Logs Policy** - Assigned to the Instance Role
- **CloudWatch Events Policy** - Assigned to the Instance Role
- **S3 Policy** - Allows all S3 functions except Delete Bucket; assigned to the Instance Role
- **Delivery Power Admin** - Assigned to the Delivery Power Admin Group
- **Delivery Read Only** - Assigned to the Delivery Read Only Group

Instance Profile

A default Instance Profile is created by the IAM template. This Instance Profile is assigned to instances created by the Gold Managed Services.

8 Creating Lambda Functions

As part of the customer on-boarding process, several Lambda functions are added to the customer account. This section describes the Lambda functions and their purpose.

.

Utility

These Lambda functions are general purpose functions and used during onboarding or when an instance is provisioned:

- **populateCustomer** - This function is created and used during the on-boarding process. It creates and populates the Customer S3 bucket. Currently, it copies the DXC Gold Managed Service components to the Customer S3 bucket to support VPC customization. Other onboarding functions might be added in the future.
- **lookupExport** - This function is used during onboarding and during instance provisioning. It looks up the value of an Export from CloudFormation stacks in a region. The name of the Export and the region are inputs to this function.
- **lookupRole** - This function is used during onboarding and during instance provisioning. The function looks up an IAM role and returns the role ARN.
- **getSOEs** - This function is used during instance provisioning. The function finds the latest version of a DXC SOE AMI. The input parameter is the name of the OS. Currently SOEs are tagged as either rhel6.7, rhel7.2 or win2012.

Managed Services

These Lambda functions are used to support the DXC Gold Managed Services:

- **backupHandler** - This function creates volume snapshots using a predefined schedule. It also removes old snapshots that have reached their retention date.
- **backupHealth** - This function checks the backup system to verify that snapshots are being created correctly.

Bastion Service

These Lambda functions are used to support the Bastion server that is used to connect to customer workloads when troubleshooting issues:

- **BastionCreateLocalAcct** - Creates a temporary user account on the target instance.
  BastionCreateIngressRuleOnTargetInstance - Creates a temporary ingress rule on the target instance.
- **BastionRandomUserAndPwd** - Creates a random user name and password.
- **BastionTremExpired** - Creates a timer event to remove the Bastion server after a specified interval.

Linux Patching Service

These Lambda functions are used to support the Linux Patching Service.

createLinuxPatchingBaseline - Creates configured *yum.conf* and *versionlock.list* files for a created policy. Creates the files to the user's specification and installs the files in the correct policy directory.

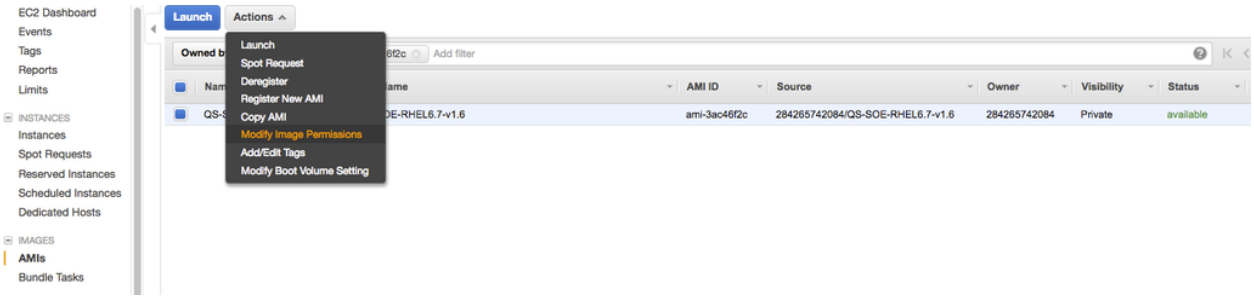
Sharing SOE AMIs

The Offering Build Team has provided a set of SOE AMIs to use when provisioning instances. AMIs must be shared into the customer account for the Simple Linux and Simple Windows templates to successfully provision.

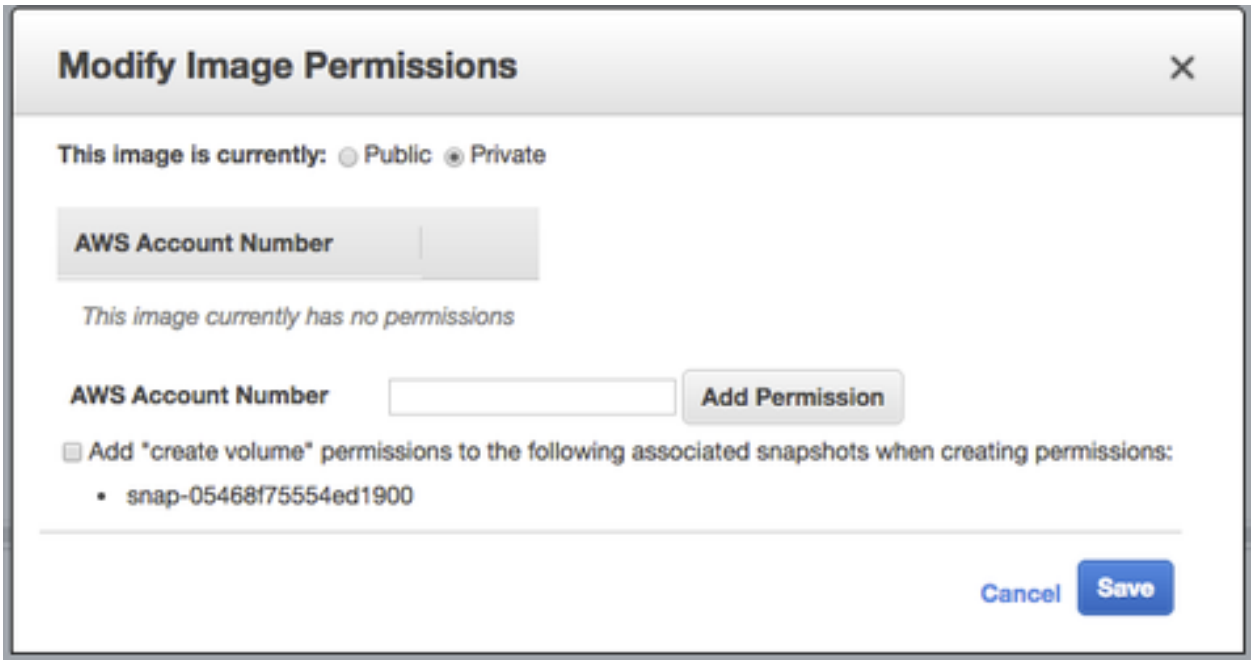
Sharing Images

To share images across AMIs into the customer account:

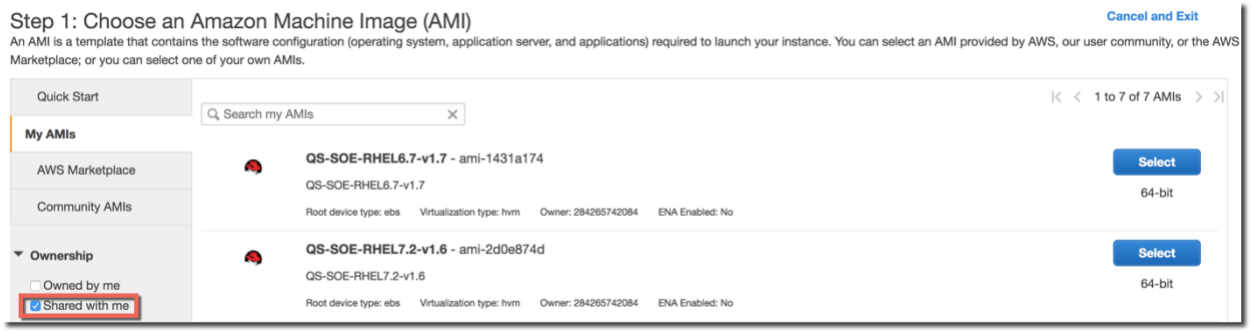
1. Log into the **PROD OBE** AWS account.
2. Switch to the desired region.
3. Under **Images**, click **AMIs** to share the images from each region to each client account.
4. Select **AMI ID**, and then from the **Actions** menu, click **Modify Image Permissions.**
   .



5. On the **Modify Image Permissions** page, type the client's **AWS Account Number** and click **Add Permission**.
   Note: You can obtain the AWS account number from an existing ARN of an IAM user.



6. Click **Save**.
7. Log out of this account or open up a different browser and log in to the Customer AWS account.
8. Go to **EC2** and click **Launch Instance**.
9. Click **My AMIs** and ensure that **Shared with me** is selected under **Ownership**.



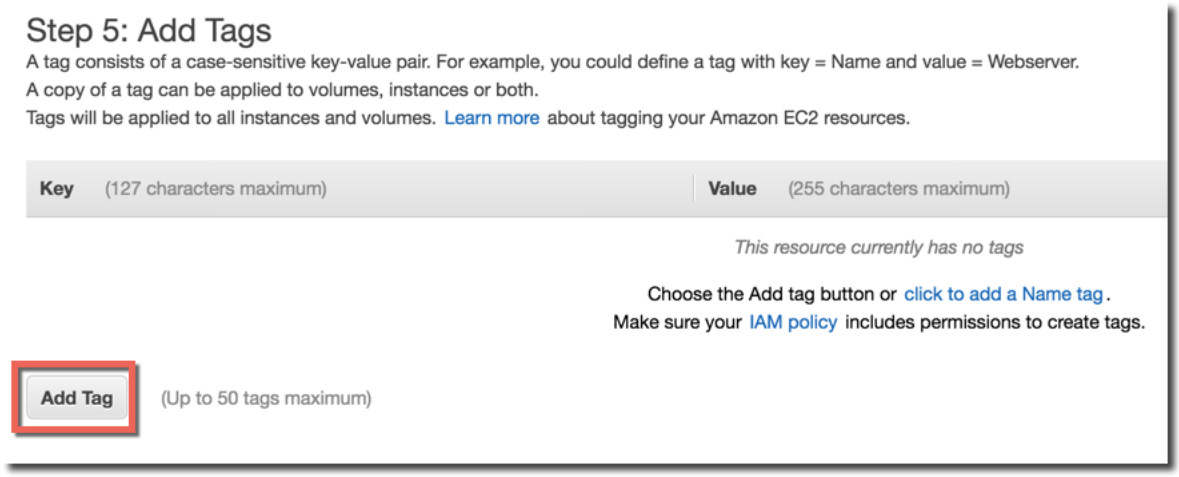
The AMI that was shared from the other account should be listed.

Note: For the AMIs to provision successfully, ensure that the tags are present after the AMI has been shared into the customer account. If they are not, add the tags to the AMI.

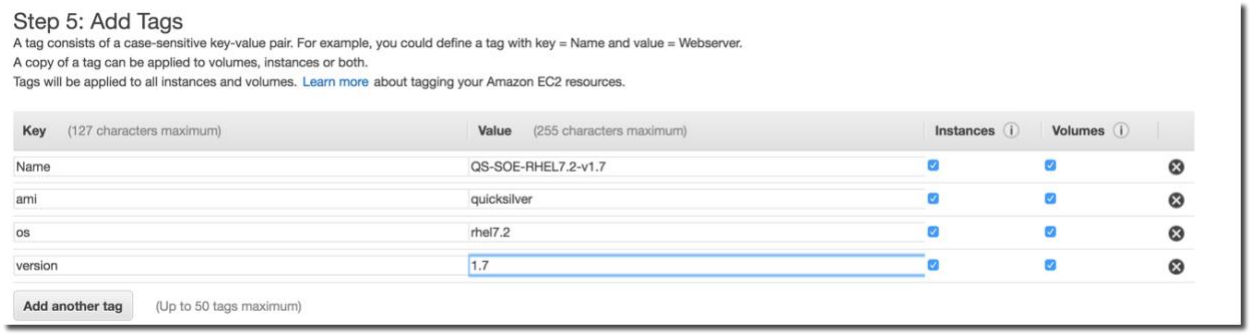
- **Name** - Full name (QS-SOE-RHEL7.2-v1.7, QS-SOE-RHEL6.7-v1.7, QS-SOE-Win2012-v1.5)
- **ami** - quicksilver
- **os** - rhel7.2, rhel6.7, or win2012, depending on the operating system
- **version** - AMI version (for example, v1.6)

To add tags:

1. Click **Select** to select the instance for which you want to add the tags.
2. Click **Next** until you get to **Step 5: Add Tags**.



3. Click **Add Tag** for each tag you are adding, and fill in a value for **Key** and **Value** for each tag.



4. When you have added all the tags, click **Next** to proceed to the next step.

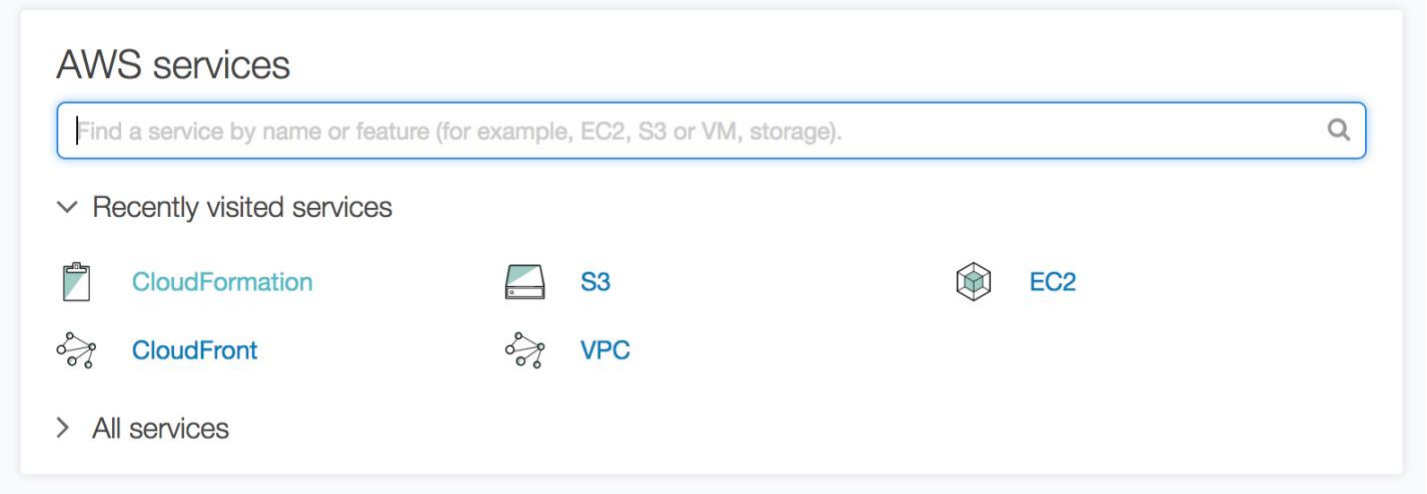
Creating a Linux AWS AMI

The creation of a Linux AWS AMI is done by executing an automation document. Then automation document is created by executing a CloudFormation Template. This stack is created as a dependent stack via the execution of the main **dxc-ms-main.yaml** CloudFormation Template. If this template has not been executed, then the following CloudFormation Template will need to be executed (assuming a customer bucket of qa.gold.dxc.obe.dev has been created) to create the following automation document:

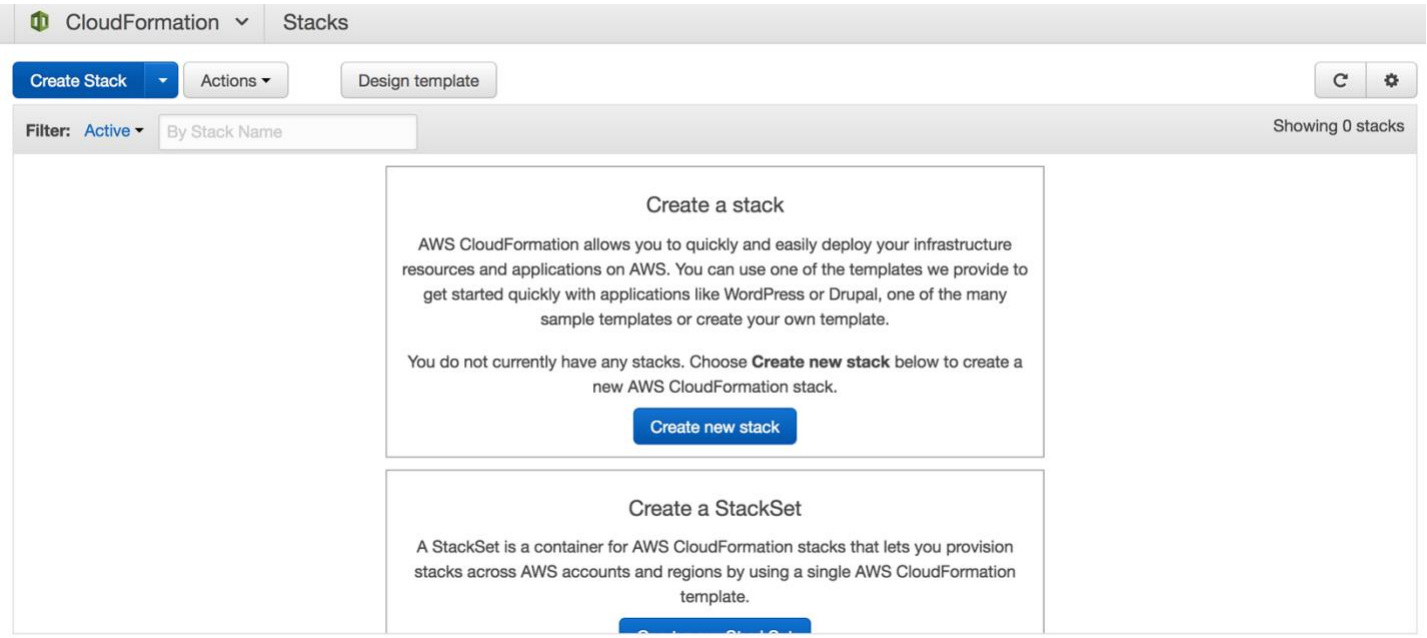
https://s3.amazonaws.com/qa.gold.dxc.obe.dev/deploy/cloudformation/QS-CreateLinuxAMI.json

To verify that the template exists:

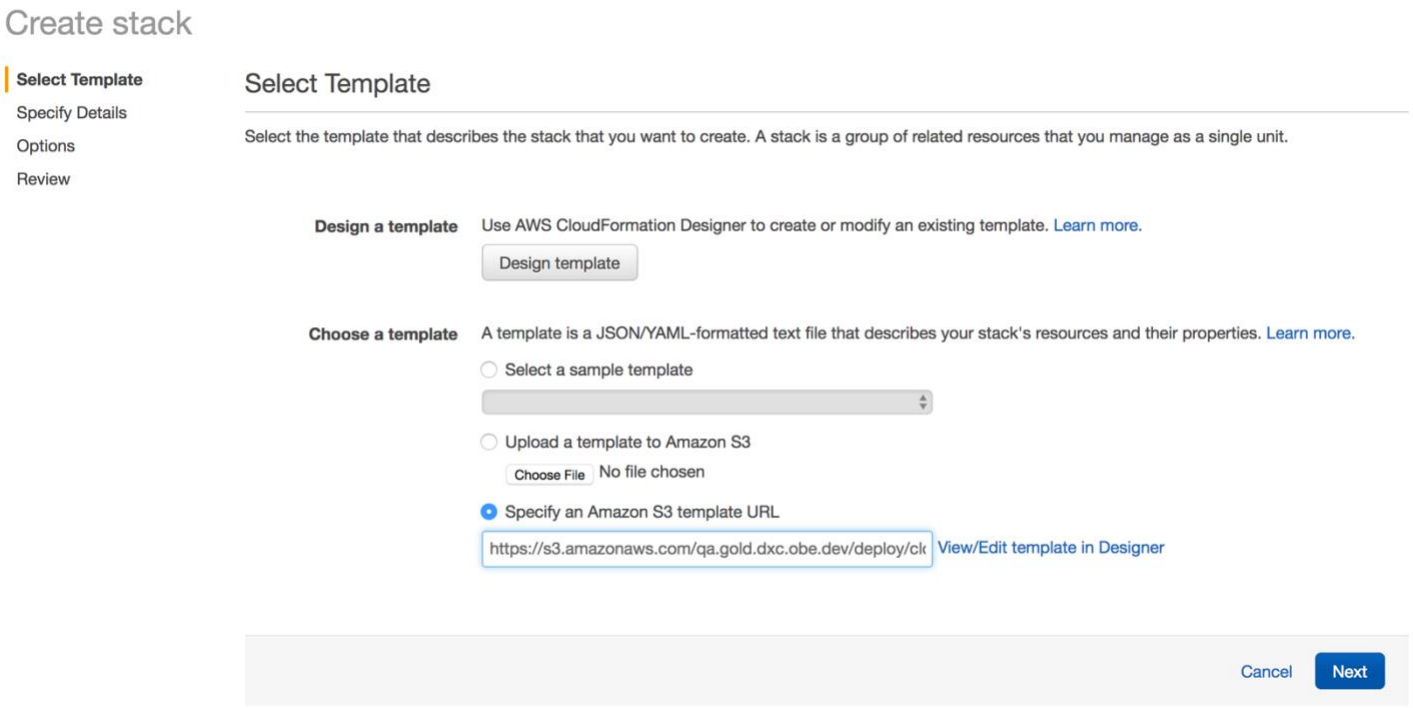
1. Log into your AWS console, and go to CloudFormation.



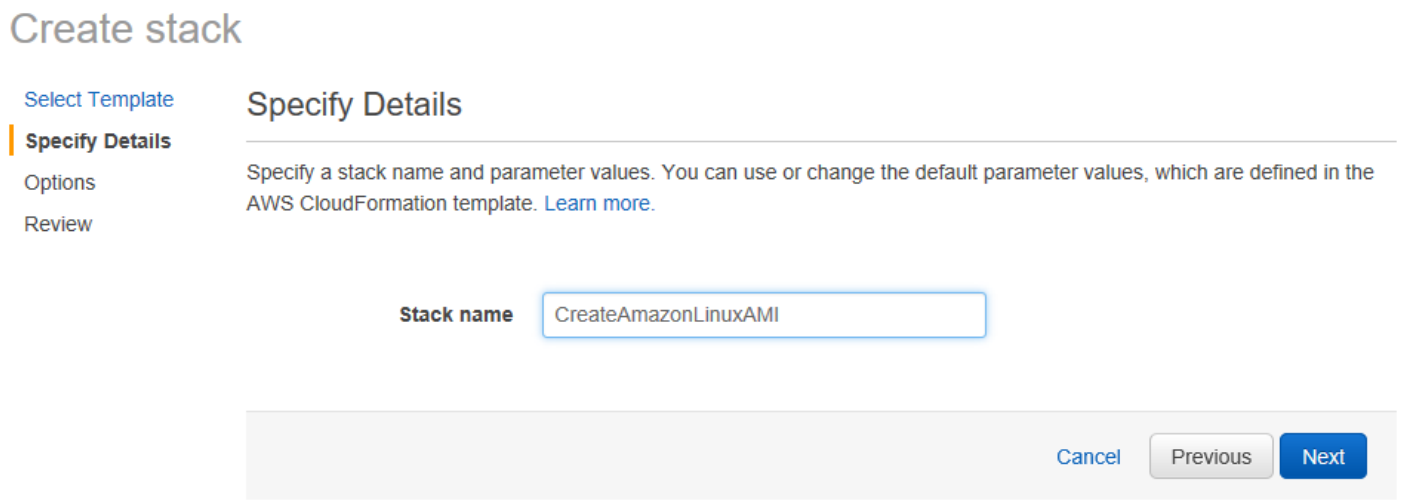
2. In CloudFormation, search for a CloudFormation template with either **CreateRhelAMIAutomation** or **CreateAmazonLinuxAMIAutomation** in its name. **This template should have already been created** by the master template (**dxc-ms-main.yaml**). **If this template exists, skip to the "Running the Automation" section.** If the template doesn't exist, continue with the next step.
3. In CloudFormation, click **Create Stack**.



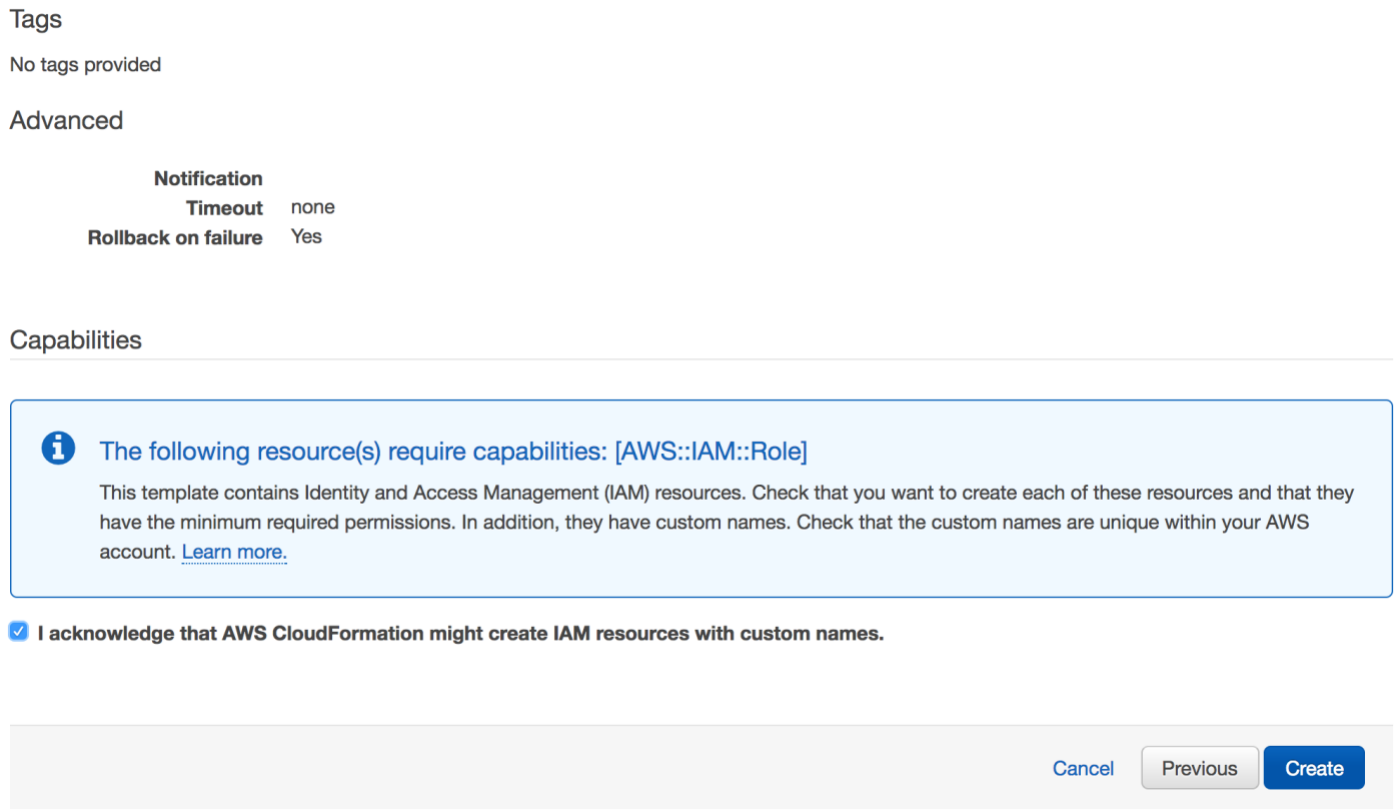
4. On the **Select Template** page, click **Specify an Amazon S3 template URL**, and paste the S3 bucket URL into the field. In our example this URL is: https://s3.amazonaws.com/qa.gold.dxc.obe.dev/deploy/cloudformation/QS-CreateSilverPlusAmazonLinuxAMI.json



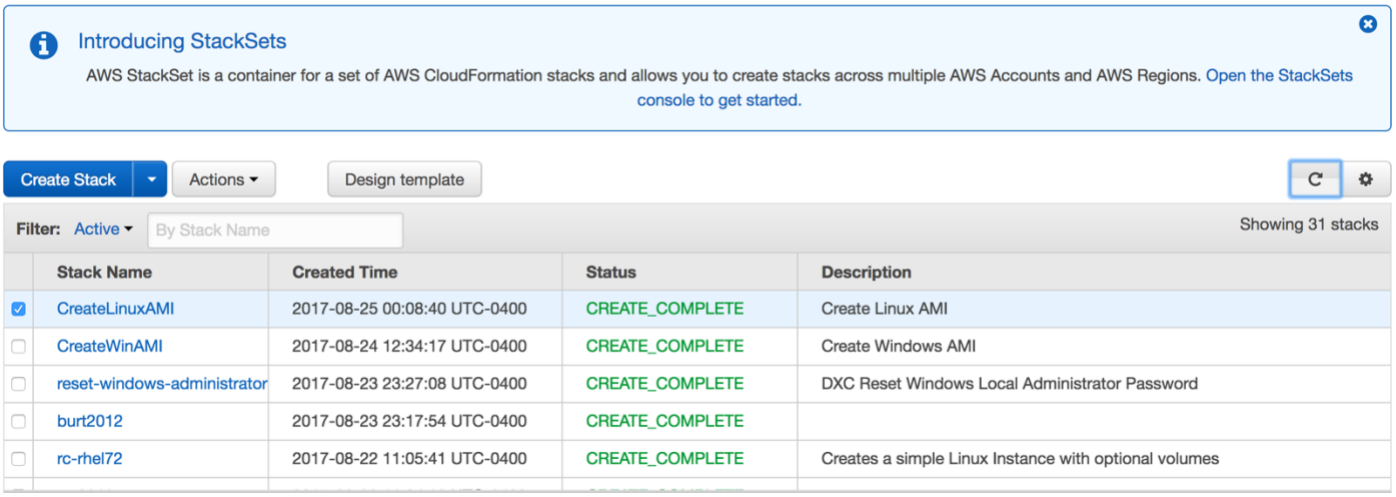
5. Click **Next**.
6. On the **Specify Details** page, in the **Stack name** field, type a descriptive name for the stack, and then click **Next**. Include either Rhel or AmazonLinux in the name, to denote which breed you're creating.



7. Click **Next** through the upcoming pages until you have reached the final page. On the final page, acknowledge the creation of IAM resources, and then click **Create**.



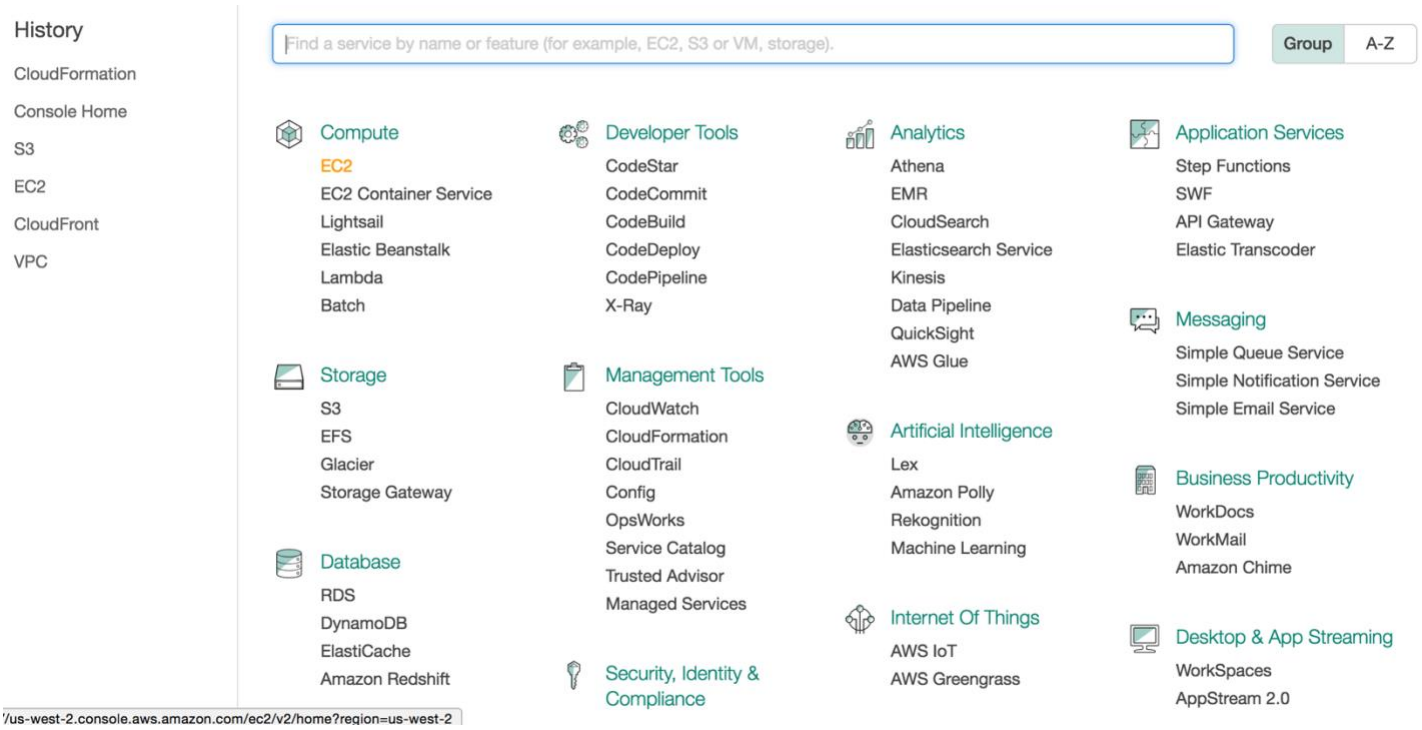
When the stack has been created you will see a **CREATE_COMPLETE** status. The automation is ready for use at this point.



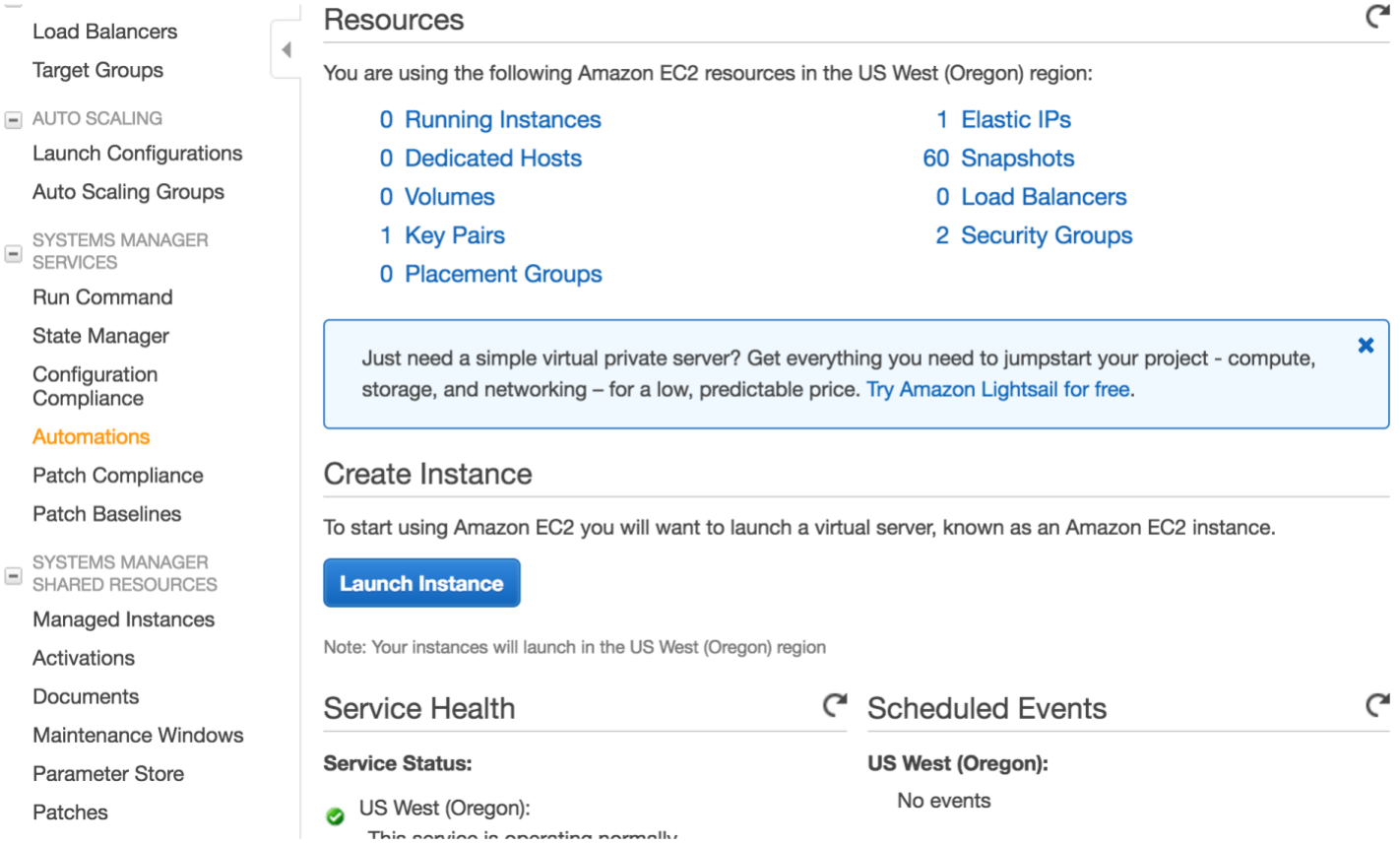
Running the Automation

To run the automation:

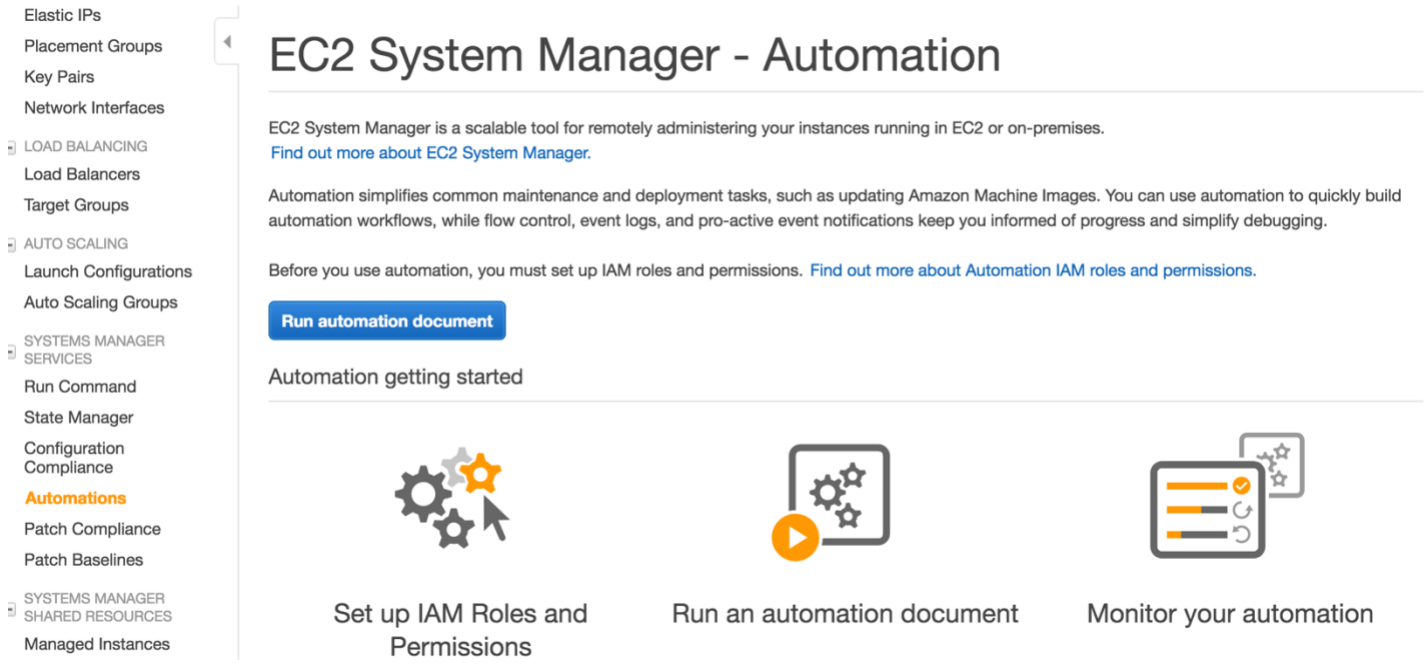
1. Log into your AWS console and go to EC2.



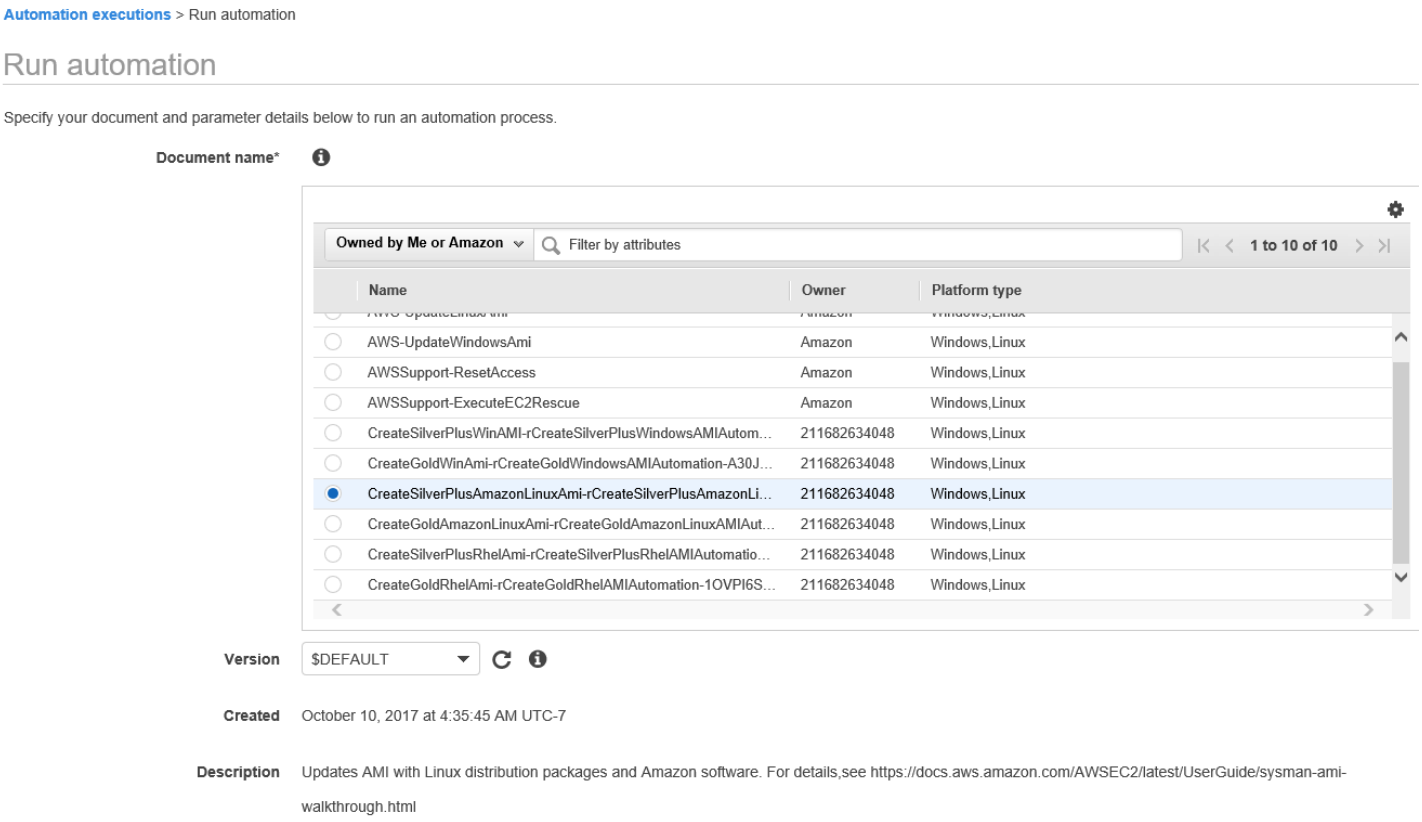
2. Under **SYSTEMS MANAGER SERVICES** in the left pane, click **Automations**.



3. On the **EC2 Systems Manager - Automation** page, click **Run automation document**.



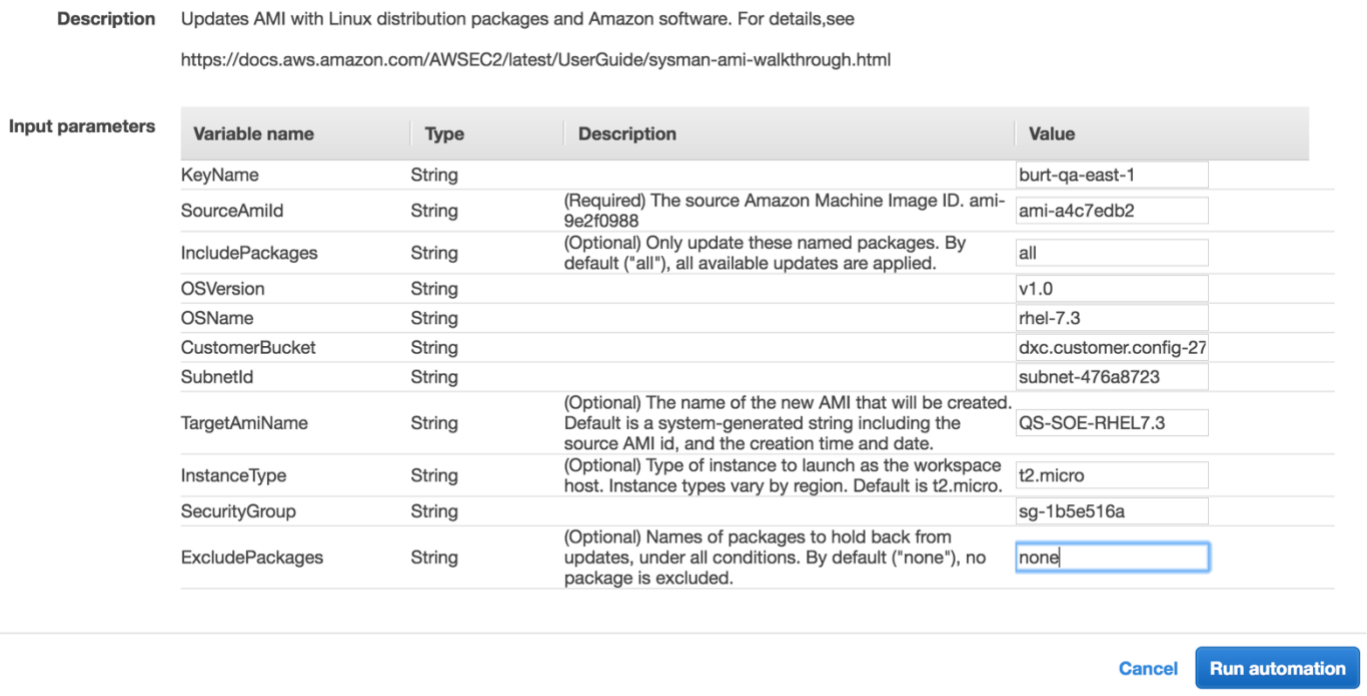
4. From the list of automation documents, select the Gold or SilverPlus Linux AMI creation automation that you created previously. Note that there are now two Linux automation documents you can execute: **CreateRhelAMIAutomation** and **CreateAmazonLinuxAMIAutomation.**



5. In the **Input parameters** section, configure the following parameters:

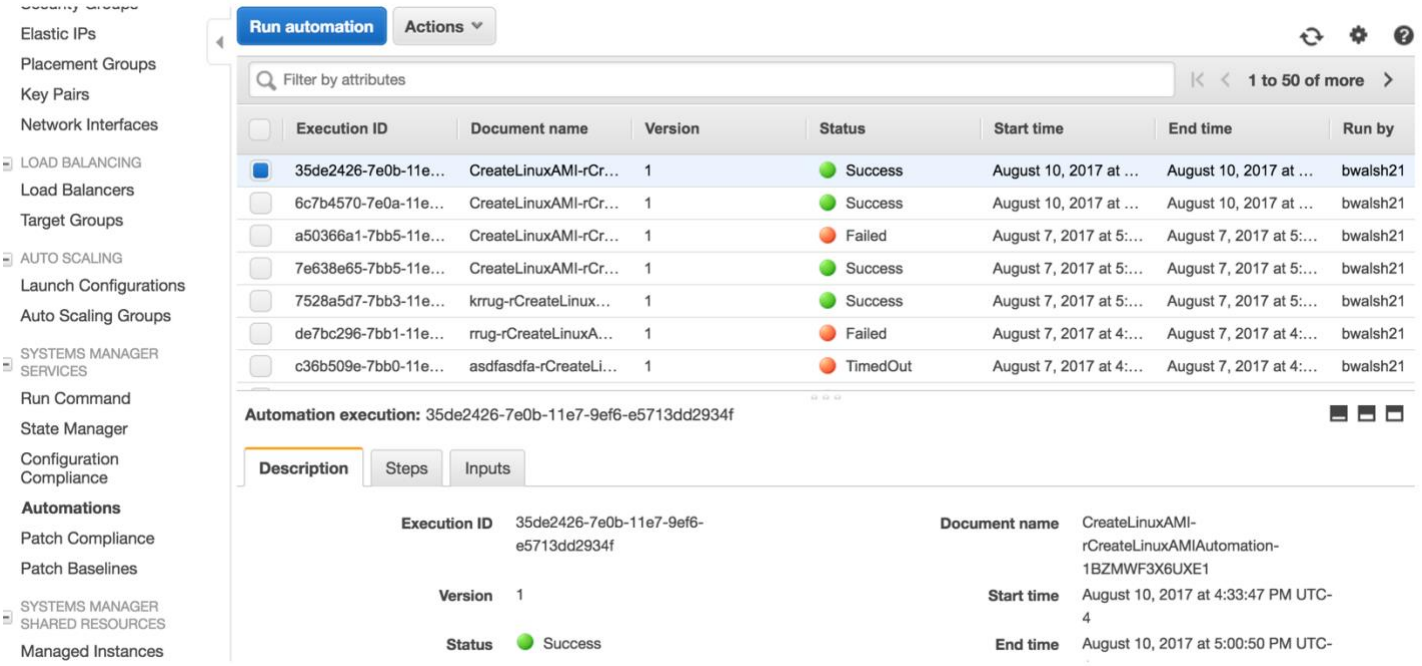
Parameter	Description
KeyName	An EC2 Key Pair in the region and account where the automation is being run.
SourceAmiId	The AMI under which the new AMI will be created. This is our starting product. The default shown is the supported AMI we use from AWS for RHEL v7.2 or Amazon Linux, depending on which automation document you execute. If you wish to create a RHEL v6.7 AMI (the only other version supported by our product), do a search in the AWS Community AMIs list for "RHEL-6.7" provided by Red Hat, Inc., and use that AMI ID for this field.
IncludePackages	The packages to be updated on the instance. Usually this is set to the default of **all**.
OSVersion	OS version. This needs to be unique under all OSName type AMIs for your region.

OSName	The name of the Linux version for which you are creating the AMI. Must be either "rhel7.2" or "rhel-6.7" for the RHEL automation, or "amazon-linux" for the Amazon Linux automation.
CustomerBucket	The name of the customer bucket from which assets will be pulled.
SubnetId	The subnet to which the Source AMI will be connected.
TargetAmiName	This is the prefix used in the creation of the AMI name for the newly created Linux AMI.
InstanceType	The EC2 Instance Type used to launch the Source AMI-created virtual machine.
SecurityGroup	The security group on the VPC associated with the SubnetId to which the instance built upon the Source AMI is connected. This security group should allow for RDP (port 3389) access.
ExcludePackages	The packages that will not be updated. Usually this is set to the default of **none**.

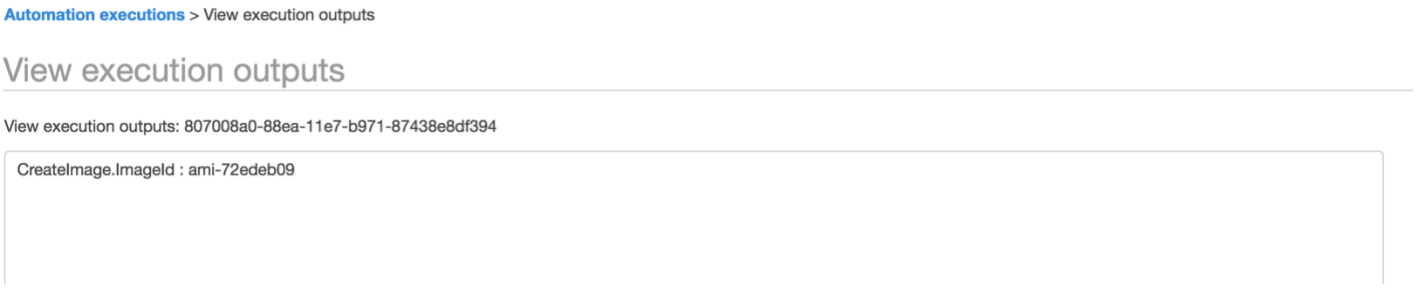


6. Click **Run automation**.

After the automation has successfully run, click the **View Output** link on the **Description** tab.



The id of the newly created AMI will be listed under **View execution outputs** from the **Actions** menu.



Creating a SUSE AWS AMI

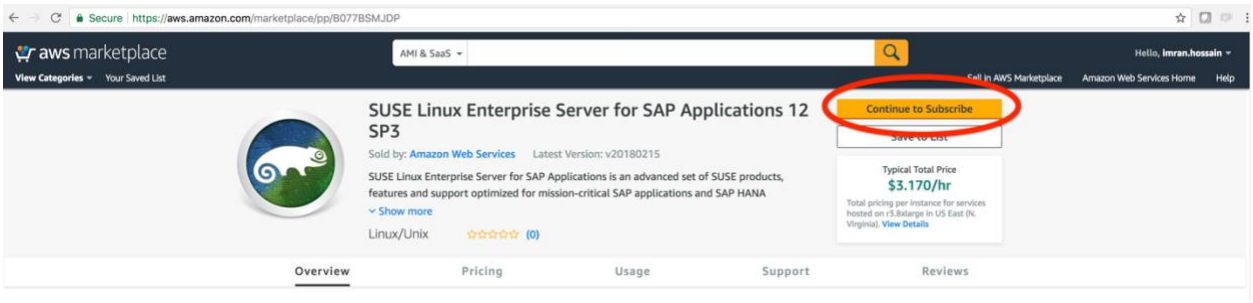
Supported AMIs

SUSE Linux Enterprise Server 12 SP3

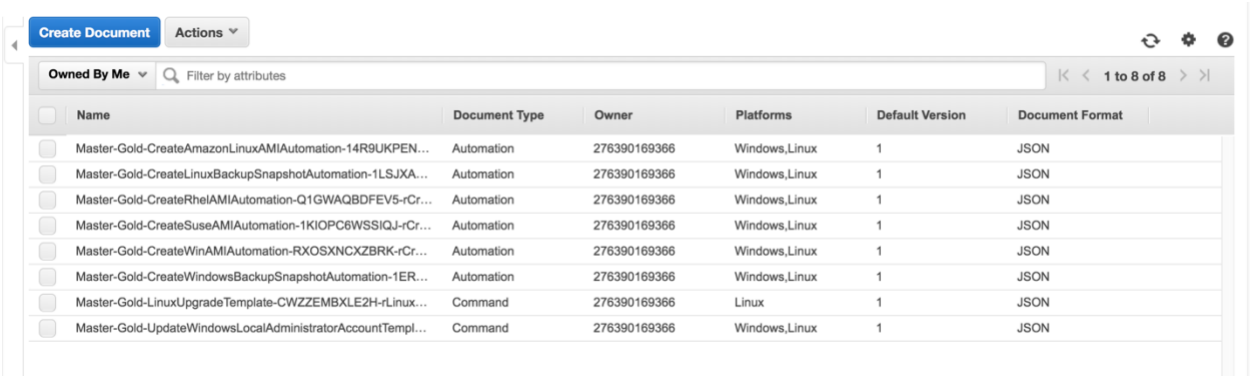
SUSE Linux Enterprise Server for SAP Applications 12 SP3

Prerequisites

AMIs from AWS Marketplace with a billing code may require user to subscribe at the Marketplace.



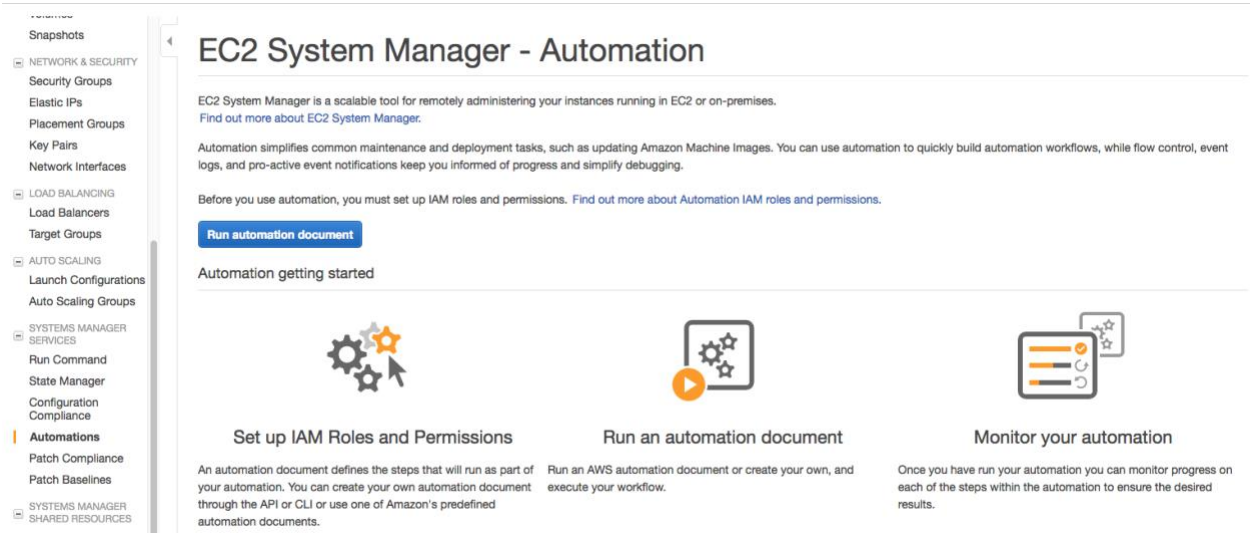
To create any one of the supported operating systems, the following Automation Documents need to be available in the region where the AMIs are to be created.



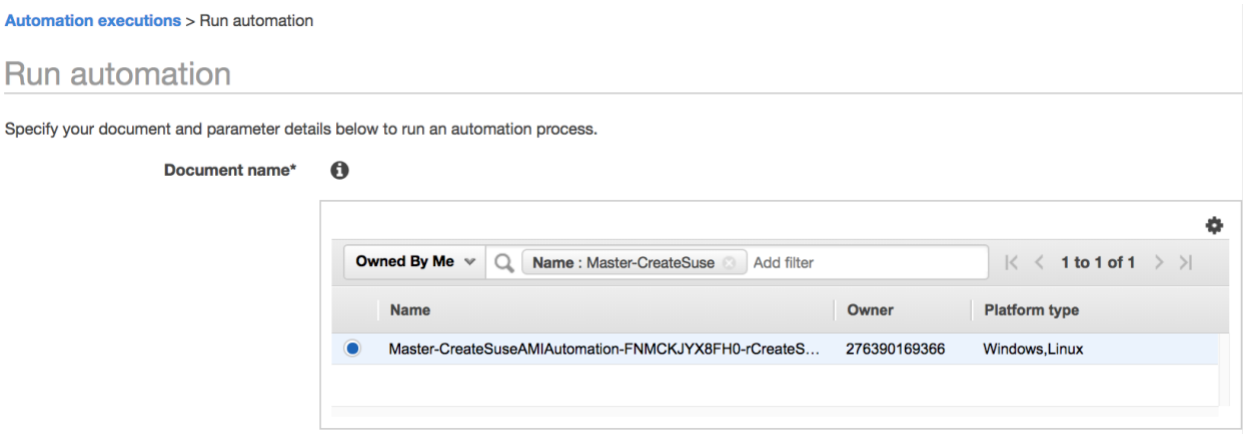
These documents are created when Master stack (*dxc-ms-main.yaml*) is created in a region.

Running the Automation

1. Under **SYSTEMS MANAGER SERVICES** in the left pane, click **Automations**.
   If this is the first time you are running automation document, you should see a splash screen like below



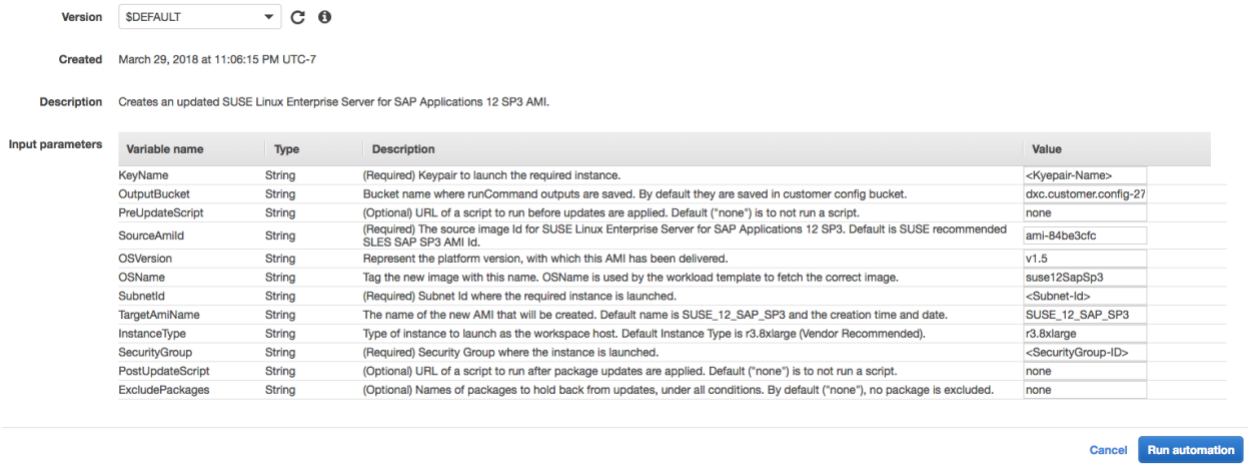
2. On the **EC2 Systems** Manager **- Automation** page, click **Run automation document**.
3. Select the automation document from the list.
   Tip: It is often easier to find the preferred document using the filters, for example "Owned By Me", "Name".



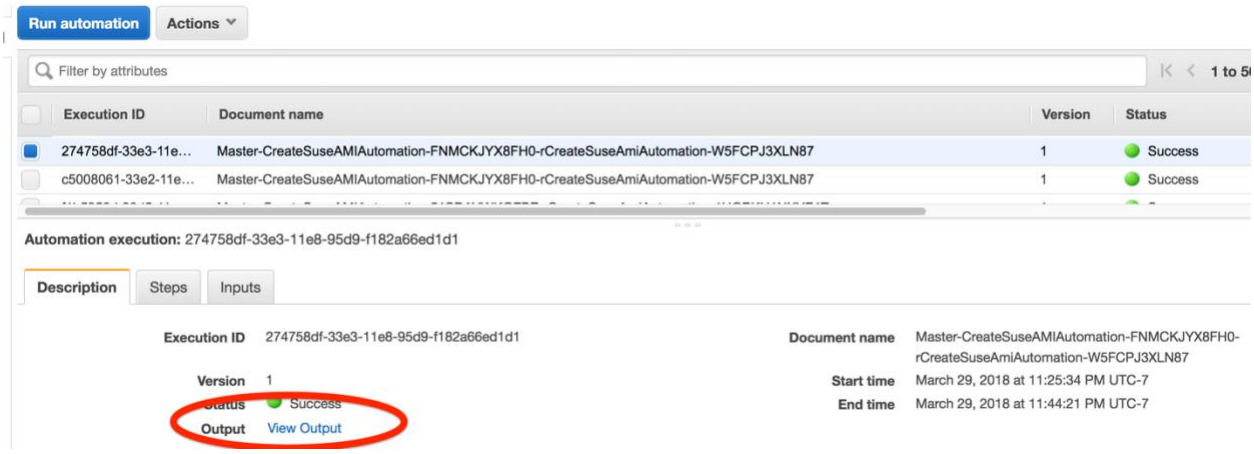
4. In the **Input parameters** section, configure the following parameters:

Parameter	Description
KeyName	An EC2 Key Pair in the region and account where the automation is being run.
OutputBucket	Bucket name where runCommand outputs are saved. By default they are saved in customer config bucket.
SourceAmiId	(Required) The source image Id for SUSE Linux Enterprise Server for SAP Applications 12 SP3. Default is SUSE recommended SLES SAP SP3 AMI Id. **Note:** Default Source AMI Id is for **SUSE Linux Enterprise Server for SAP Applications 12 SP3**. If you are using **SUSE Linux Enterprise Server 12 SP3**, this AMI Id should be an AWS provided SLES 12 SP3 AMI Id.
IncludePackages	The packages to be updated on the instance. Usually this is set to the default of**all**.
OSVersion	Represent the platform version, with which this AMI has been delivered.
OSName	Tag the new image with this name. OSName is used by the workload template to fetch the correct image. Allowed tags for SLES images are: **sles12-sp3-sap** - SUSE Linux Enterprise Server for SAP Applications 12 SP3 **sles12-sp3** - SUSE Linux Enterprise Server 12 SP3
SubnetId	(Required) Subnet Id where the required instance is launched.
TargetAmiName	The name of the new AMI that will be created. Default name is DXC_SUSE_12_SP3_SAP and the creation time and date.
InstanceType	Type of instance to launch as the workspace host. Default Instance Type is r3.8xlarge (Vendor Recommended) for SLES SP3 SAP image. **Note:** If you are using **SUSE Linux Enterprise Server 12 SP3**, instance type can be greater or equal to t2.small.

SecurityGroup	(Required) Security Group where the instance is launched. Security group should allow port 22 for Linux images.
PreUpdateScript	(Optional) URL of a script to run before updates are applied. Default ("none") is to not run a script.
PostUpdateScript	(Optional) URL of a script to run after package updates are applied. Default ("none") is to not run a script.
ExcludePackages	(Optional) Names of packages to hold back from updates, under all conditions. By default ("none"), no package is excluded.

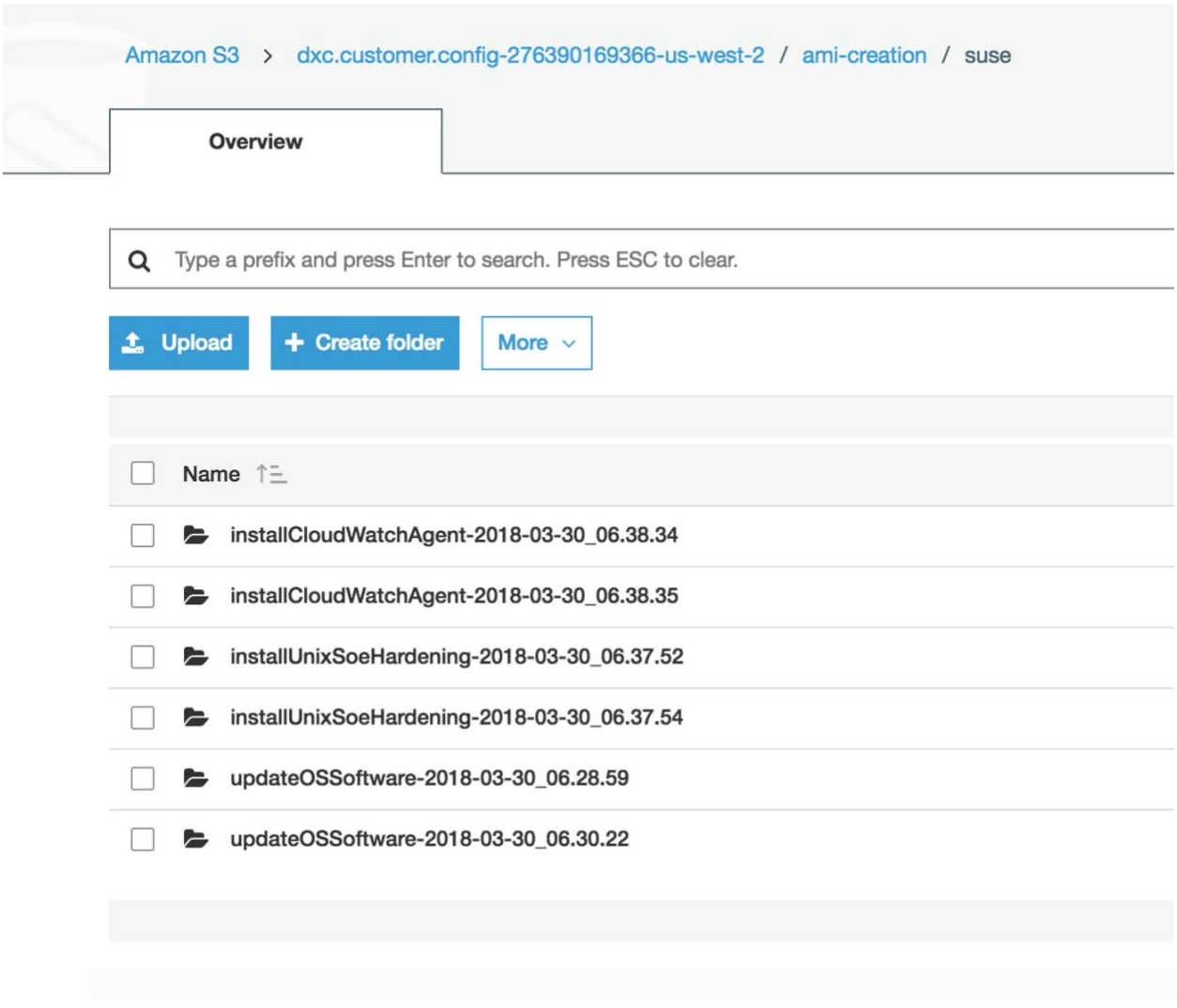


5. Click **Run** automation.
6. After the automation has successfully completes, click the View **Output** link on the **Description** tab.



The id of the newly created AMI will be listed under **View execution outputs** from the **Actions** menu.

Automation steps with aws:runCommand Action Type stores the outputs to S3 bucket that is provided as OutputBucket, example location, <bucketname>/ami-creation/suse/



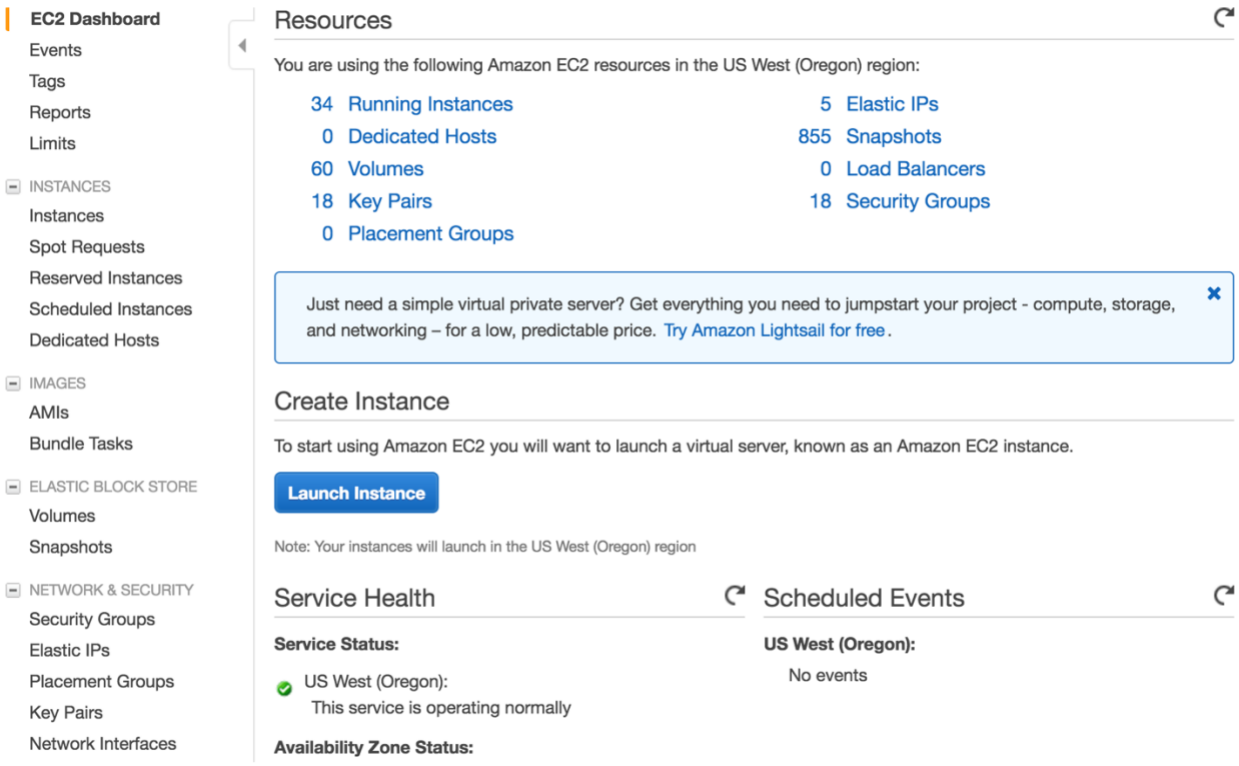
Creating a Windows AWS AMI

This section describes how to use the EC2 Automation to generate a Windows 2012 R2 or Windows 2016 AMI.

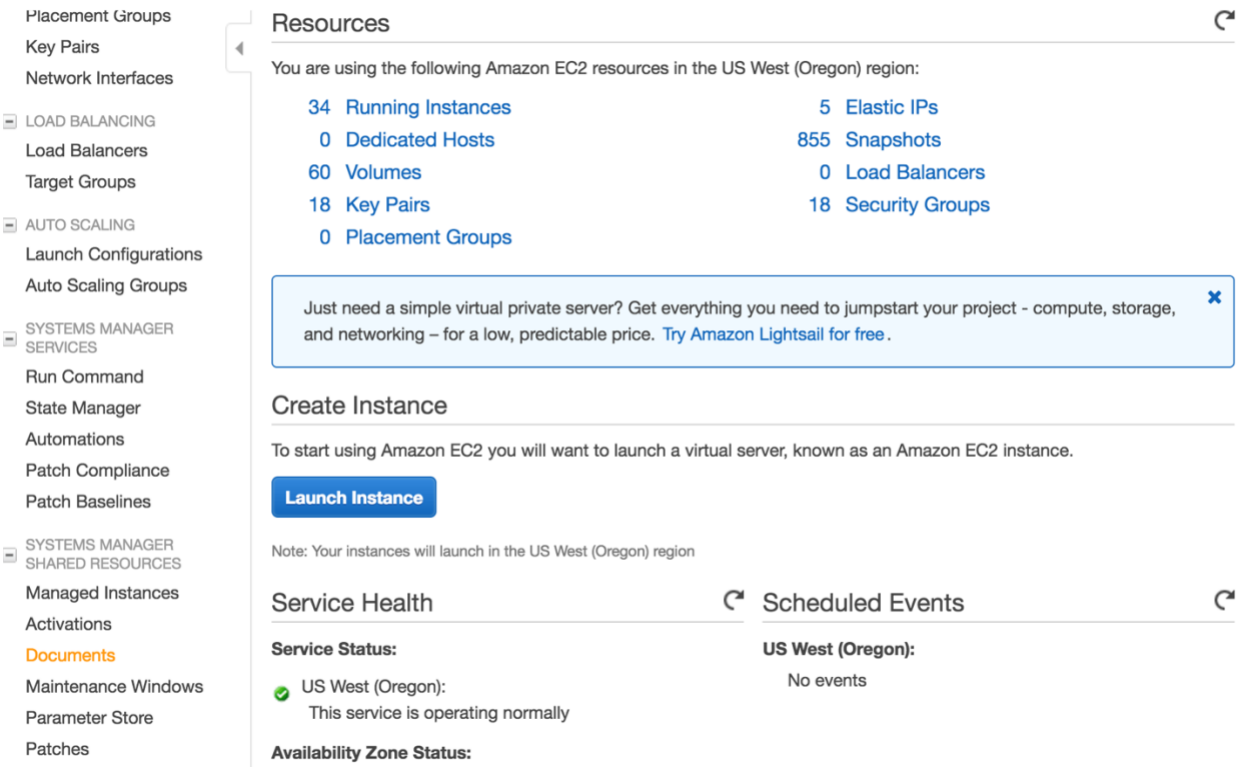
Installing the Solution

You run an AWS Command Automation script to create the AMI. This script gets created in AWS by running a CloudFormation template, which is referenced in the master *dxc-ms-main.yaml* CloudFormation template. If you run this template, the automation will be present.

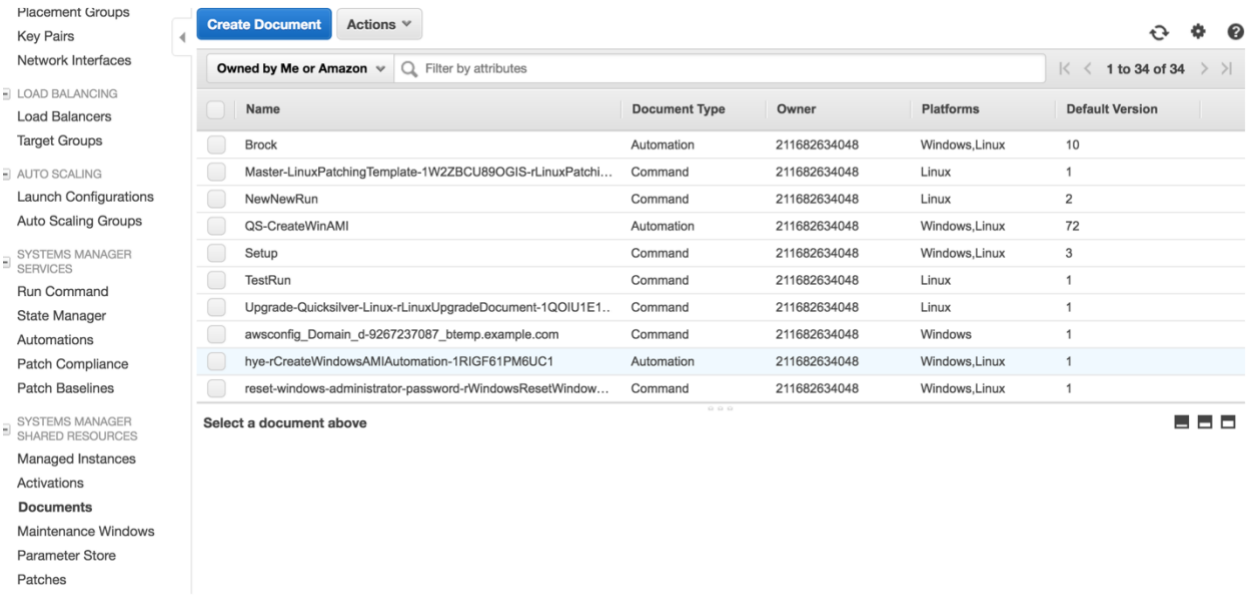
1. To test if the Command Automation script has been installed, navigate to **EC2**.



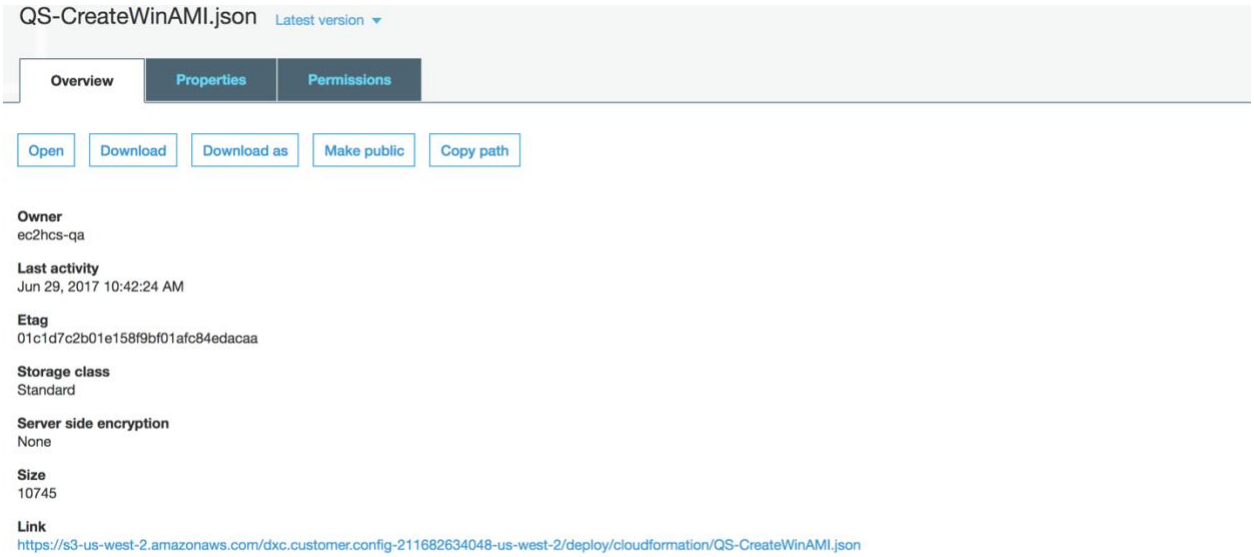
2. Under **SYSTEMS MANAGER SHARED RESOURCES**, click **Documents**.



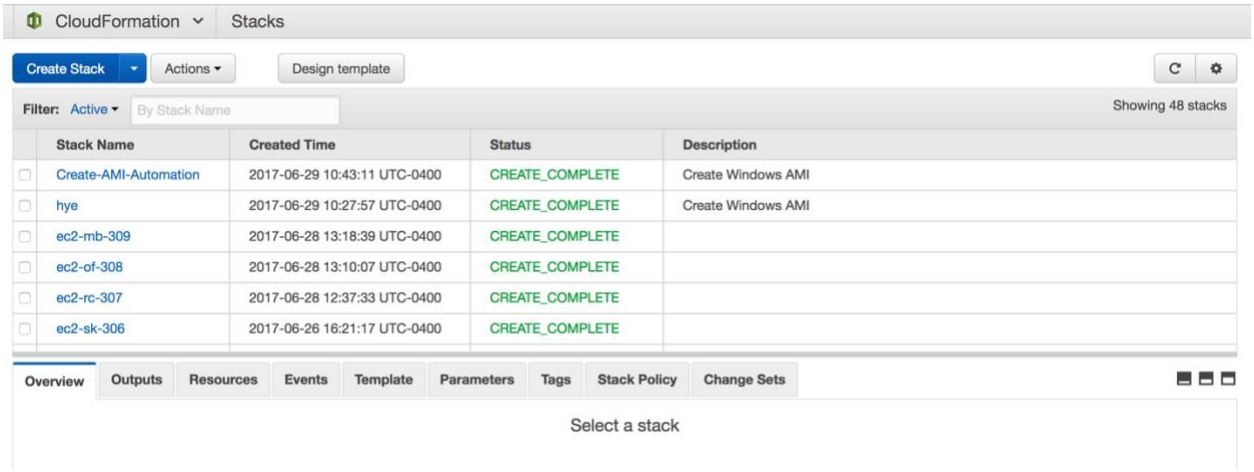
3. Search for an Automation Document Type with part of the name *rCreateWindowsAMI*.
   In the example, there is a script called *(hye-rCreateWindowsAMIAutomation-1RIGF61PM6UC1)*.



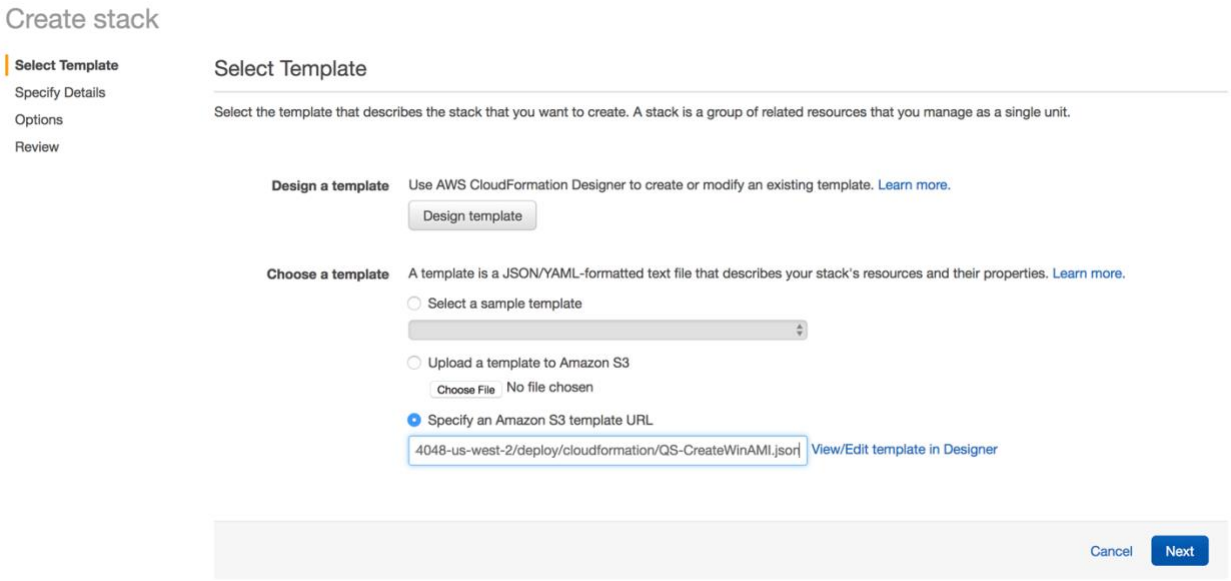
4. If the Command Automation script does not exist, run the *QS-CreateWinAMI.json* CloudFormation template to create the Command Automation script.
   Find the QS-CreateWinAMI.json CloudFormation template in S3 or on your local file system. In the example, the template is in S3 at the following URL:
   ttps://s3-us-west-2.amazonaws.com/dxc.customer.config-211682634048-us-west-2/deploy/cloudformation/QS-CreateWinAMI.json



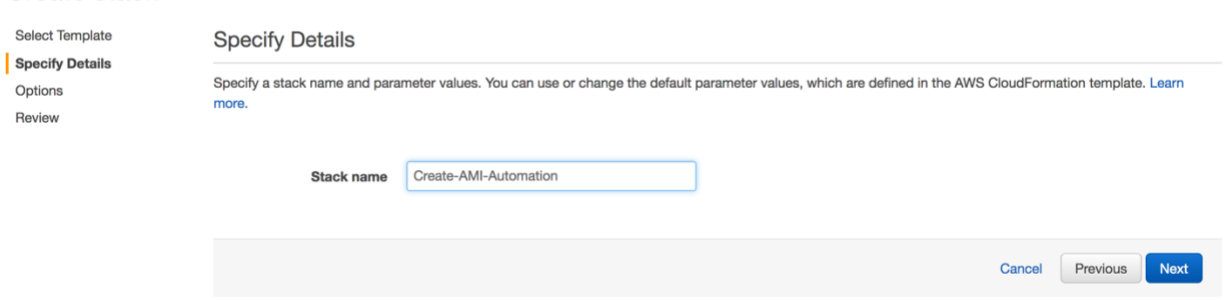
5. Go to CloudFormation in AWS and click **Create Stack**.



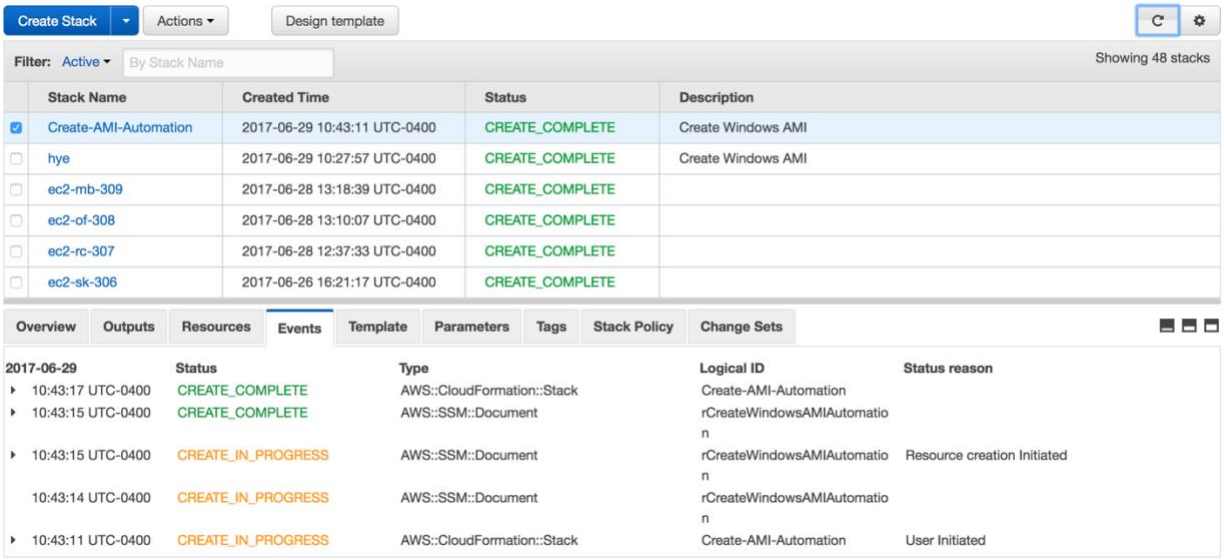
6. Paste the S3 CloudFormation template URL into the **Choose a template** field.



7. Click **Next**.
8. Type a name in the **Stack name** box and click **Next**.
9. Click **Next** again and then click **Create**.



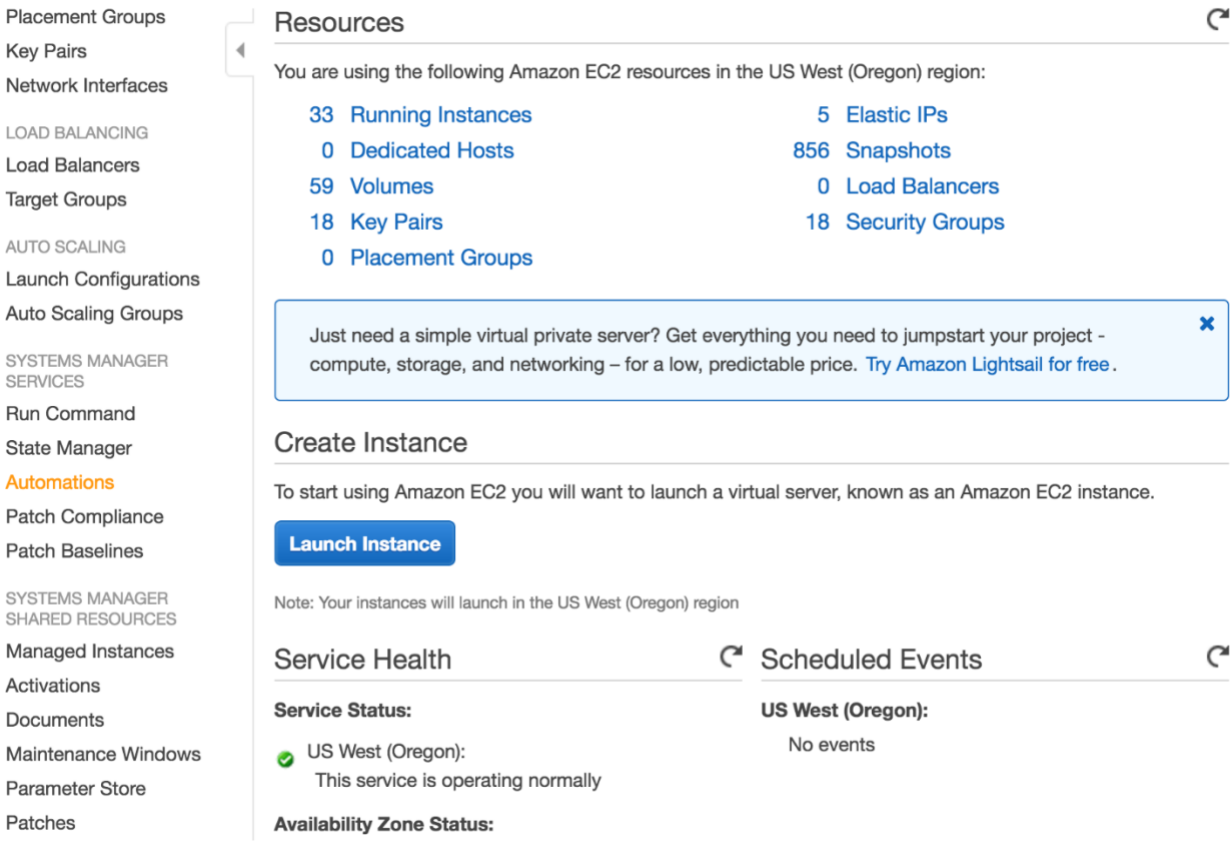
10. When the stack is created, the automation is in place.



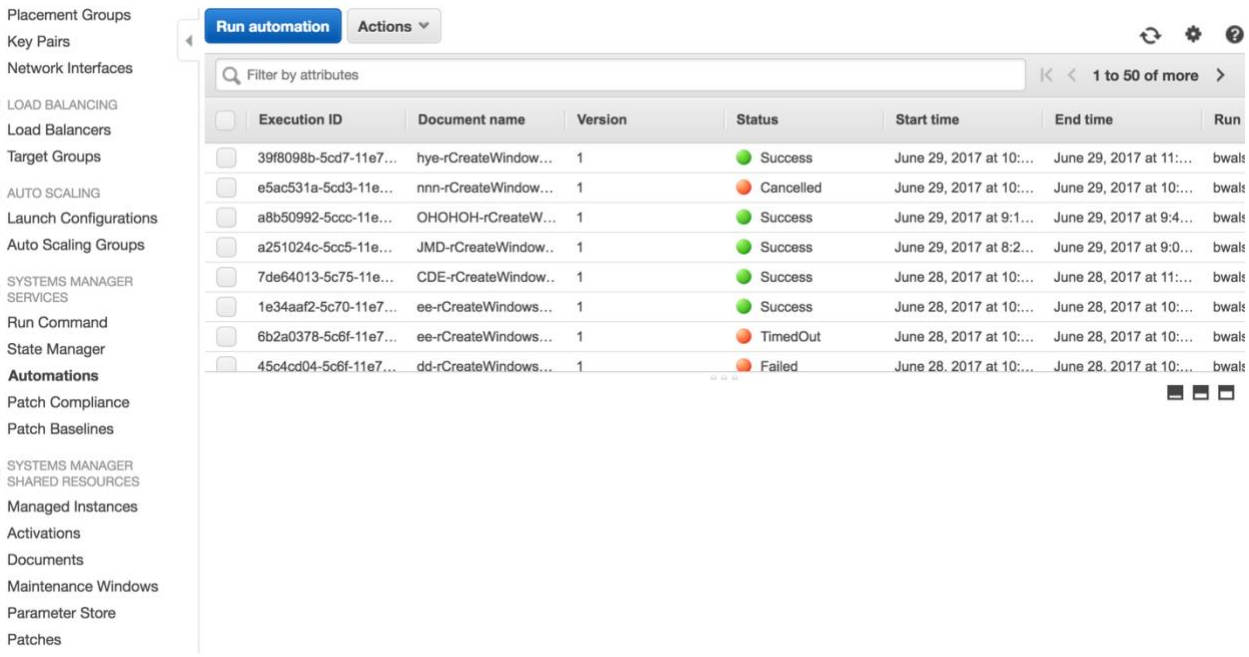
Running the Command Script

To run the Command Automation Script:

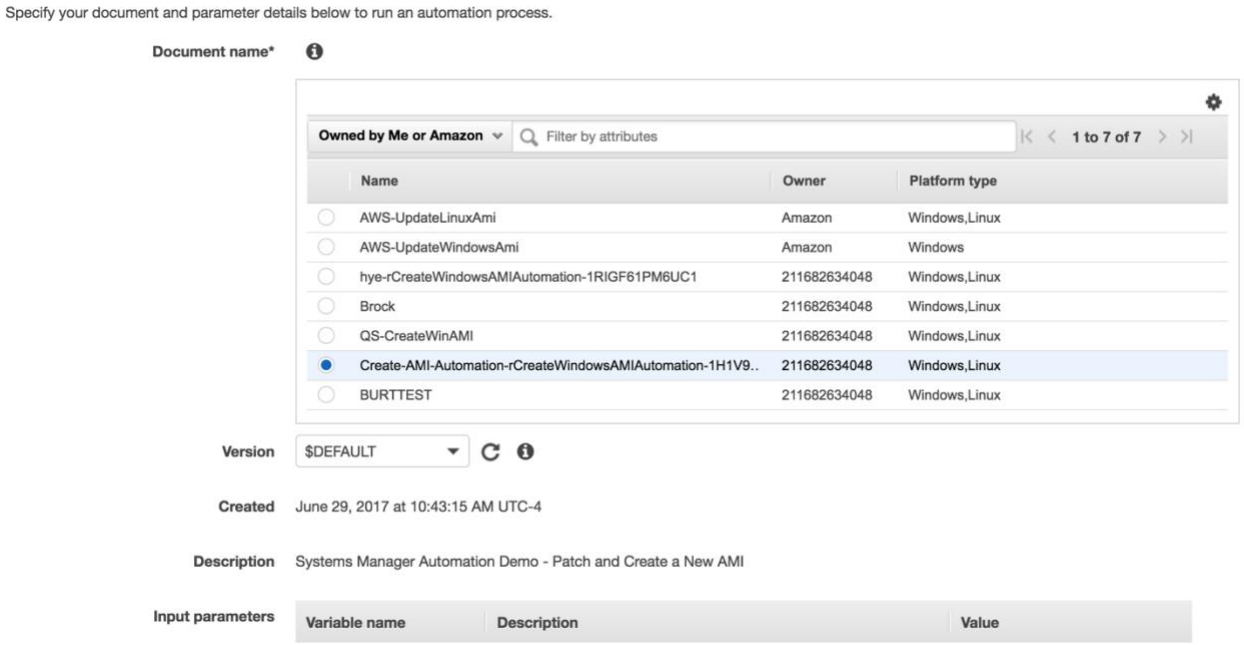
1. Under EC2, click **Automations** on the left.



2. Click **Run Automation**.



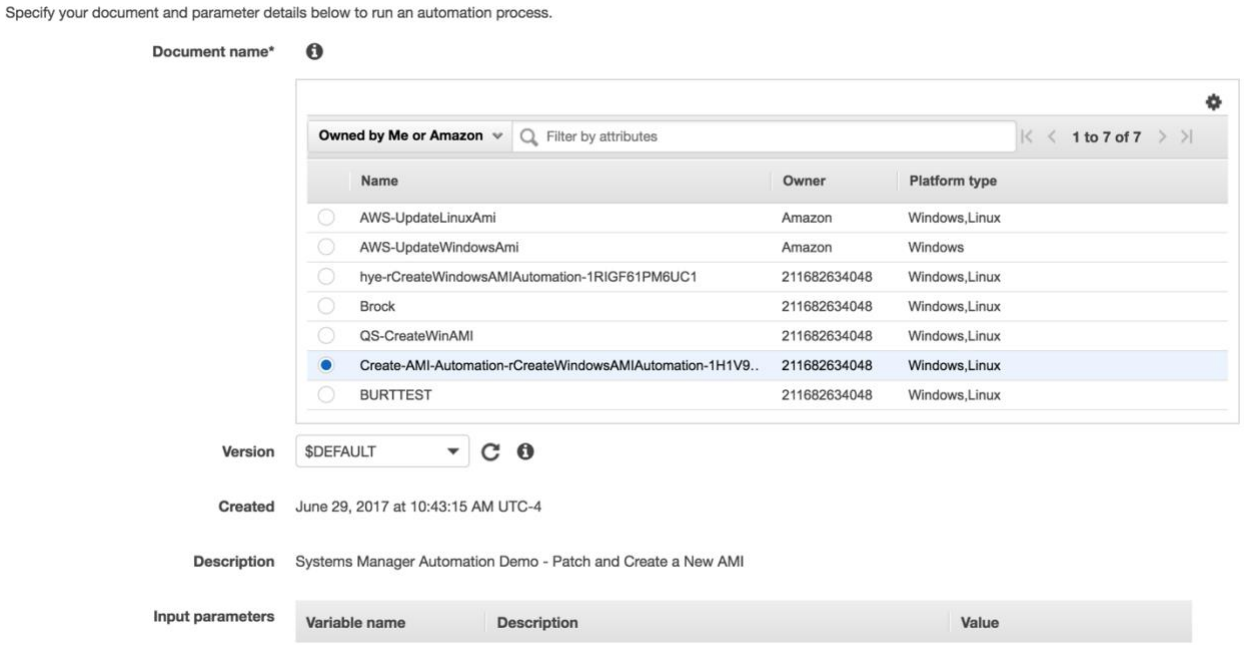
3. Choose the Command Automation document that was created by running the CloudFormation template (if it did not already exist) above.



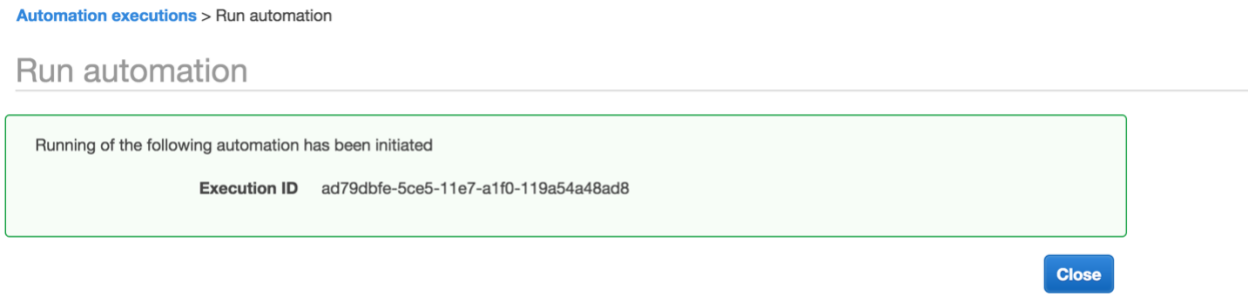
4. In the **Input parameters** section, configure the following parameters:

Parameter	Description
KeyName	An EC2 Key Pair in the region and account where the automation is being run.
CrowdStrikeCID	The key value for the CrowdStrike software being loaded on the AMI.

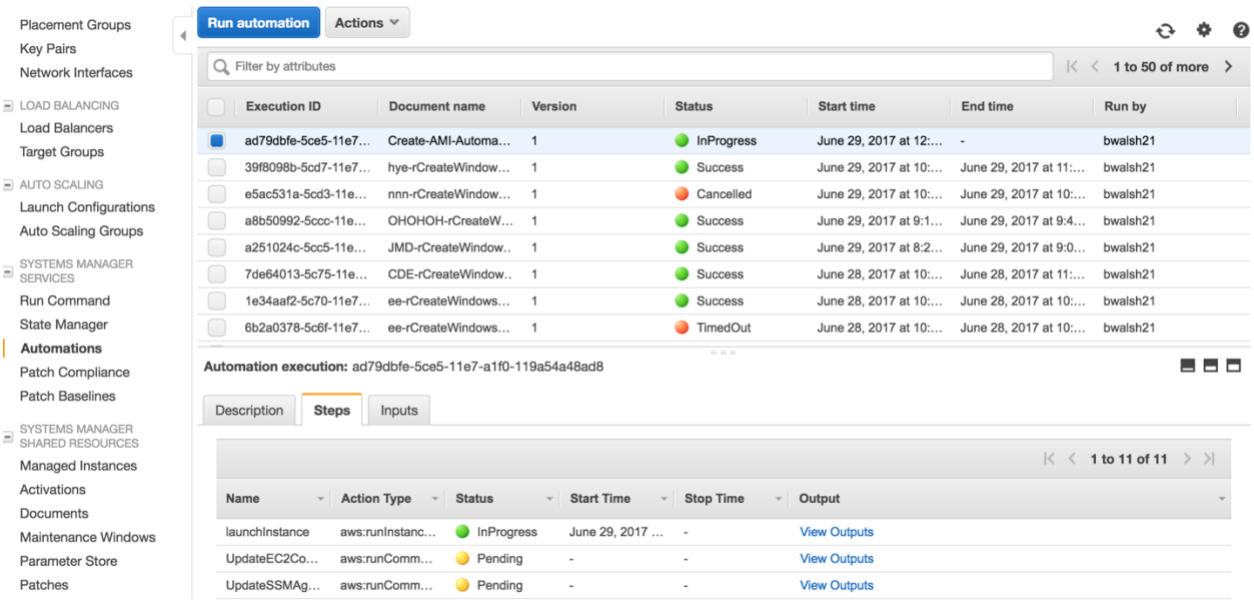
SourceAmiId	The AMI under which the new AMI will be created. This is our starting product. The default shown is the supported AMI we use from AWS for Windows 2016. If you wish to create a Windows 2012 R2 AMI (the only other version supported by our product), do a search in Quick Start list of AWS AMIs for "2012" and use the AMI ID for "Microsoft Windows Server 2012 R2 Base" for this field.
OSVersion	OS version. This needs to be unique under all OSName type AMIs for your region.
OSName	The name of the Windows version for which you are creating the AMI. Must be either "win2016" or "win2012".
CustomerBucket	The name of the customer bucket from which assets will be pulled.
SubnetId	The subnet to which the Source AMI will be connected.
TargetAmiName	This is the prefix used in the creation of the AMI name for the newly created Windows AMI.
InstanceType	The EC2 Instance Type used to launch the Source AMI-created virtual machine.
SecurityGroup	The security group on the VPC associated with the SubnetId to which the instance built upon the Source AMI is connected. This security group should allow for RDP (port 3389) access.



5. Click **Run automation**.
6. Click **Close.**



7. Click the button in the first column of the execution and then the **Steps** tab in the bottom window to follow the execution.



8. After the execution completes, click **Description** and then **View Output**.

You should see the created AMI ID.

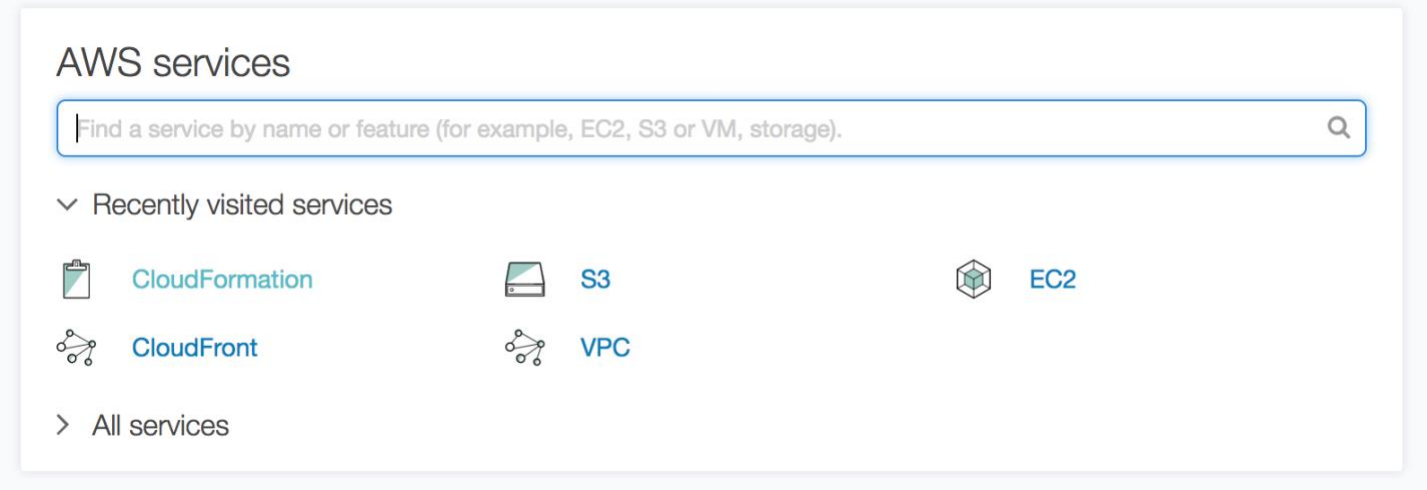
Creating a Windows AWS AMI for Silver Plus Customers

The creation of a Windows AWS AMI is done by executing an automation document. The automation document is created by executing a CloudFormation Template. This stack is created as a dependent stack via the execution of the main **dxc-ms-main.yaml** CloudFormation Template. If this template has not been executed, then the following CloudFormationTemplate will need to be executed (assuming a customer bucket of **qa.gold.dxc.obe.dev** has been created) to create the following automation document:

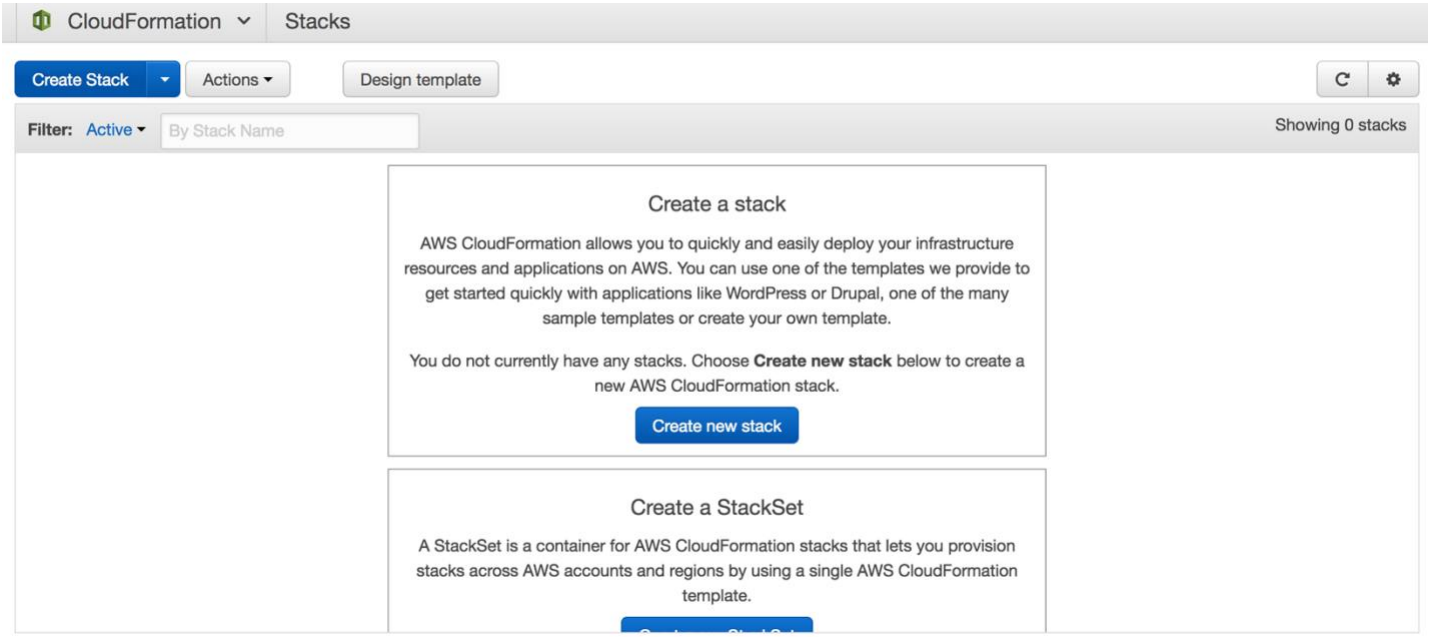
https://s3.amazonaws.com/silver.dxc.obe.dev/deploy/cloudformation/QS-CreateWinAMISilverPlus.json

To verify that the template exists:

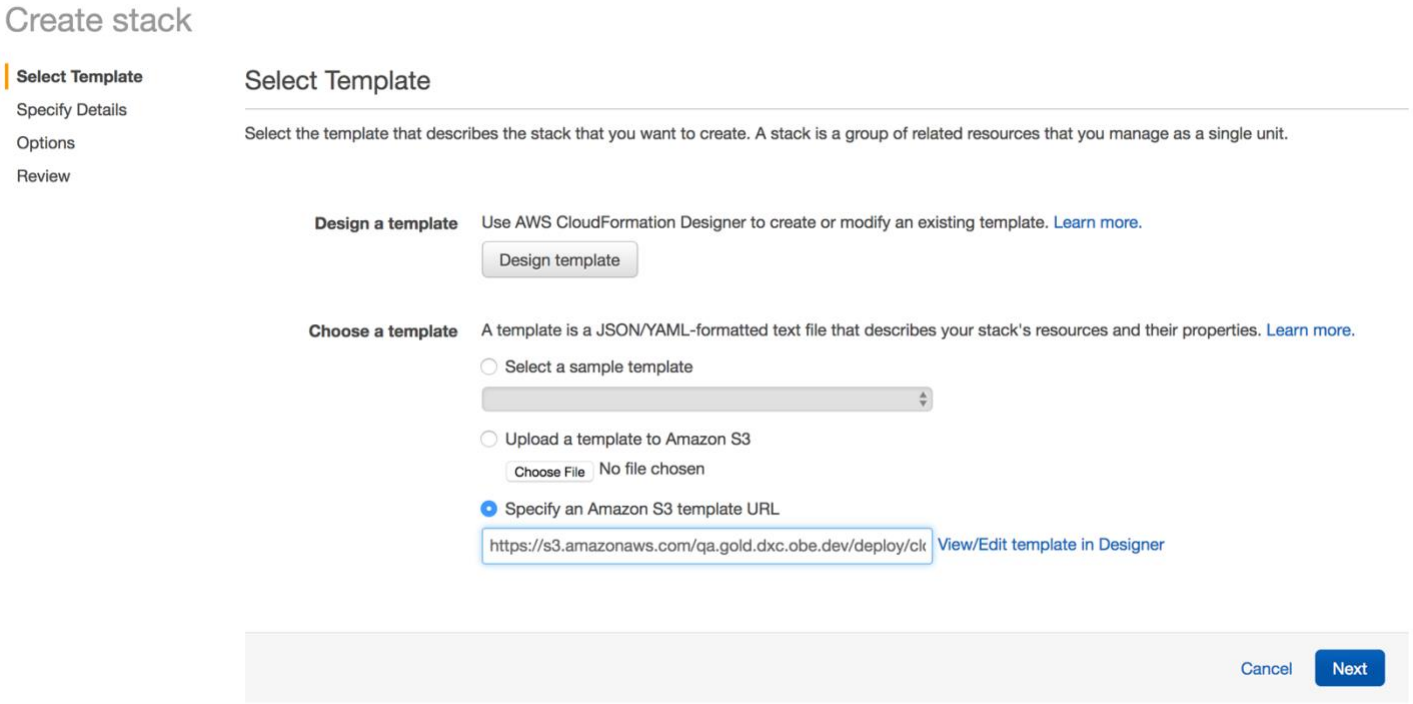
1. Log into your AWS console, and go to CloudFormation.



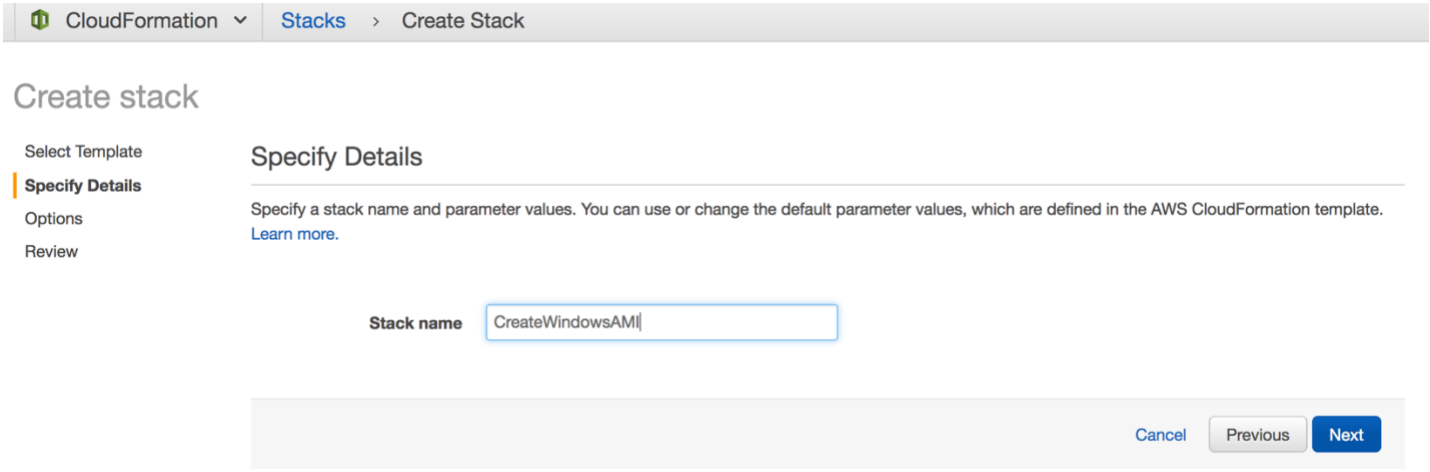
2. In CloudFormation, search for a CloudFormation template with **CreateAMITemplate** (release 1.0) or **CreateWinAMIAutomation** (after release 1.0) in its name. If this template exists, skip to the "Running the Automation" section. If the template doesn't exist, continue with the next step.
3. In CloudFormation, click **Create Stack**.



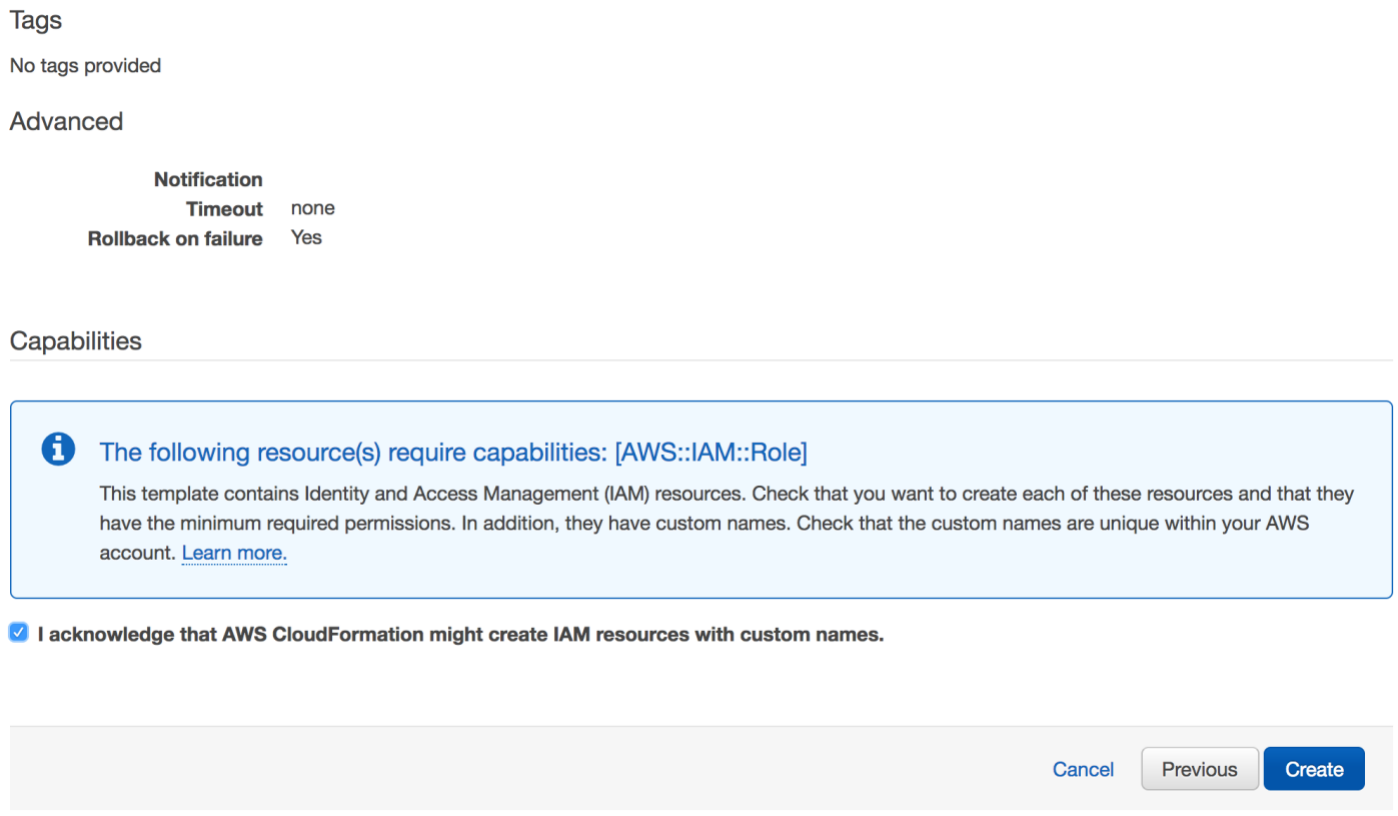
4. On the **Select Template** page, click **Specify an Amazon S3 template URL**, and paste the S3 bucket URL into the field. In our example, this URL is https://s3.amazonaws.com/silver.dxc.obe.dev/deploy/cloudformation/QS-CreateWinAMISilverPlus.json.



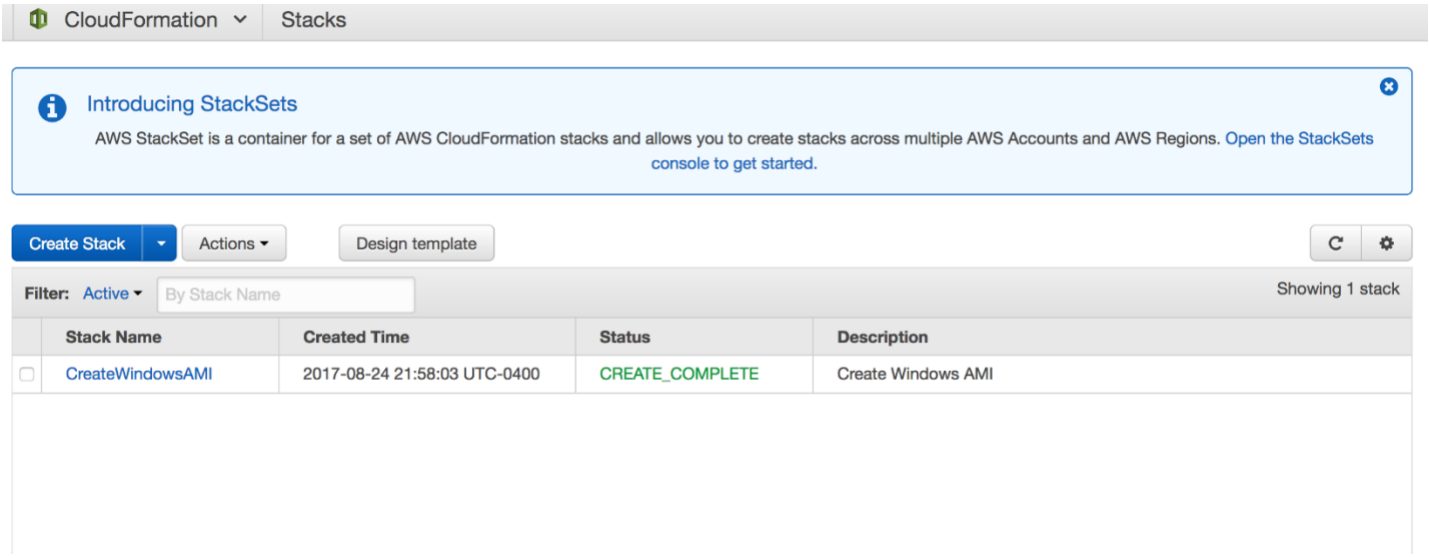
5. Click **Next**.
6. On the **Specify Details** page, in the **Stack name** field, type a descriptive name for the stack, and then click **Next**.



7. Click **Next** through the upcoming pages until you have reached the final page. On the final page, acknowledge the creation of IAM resources, and then click **Create**.



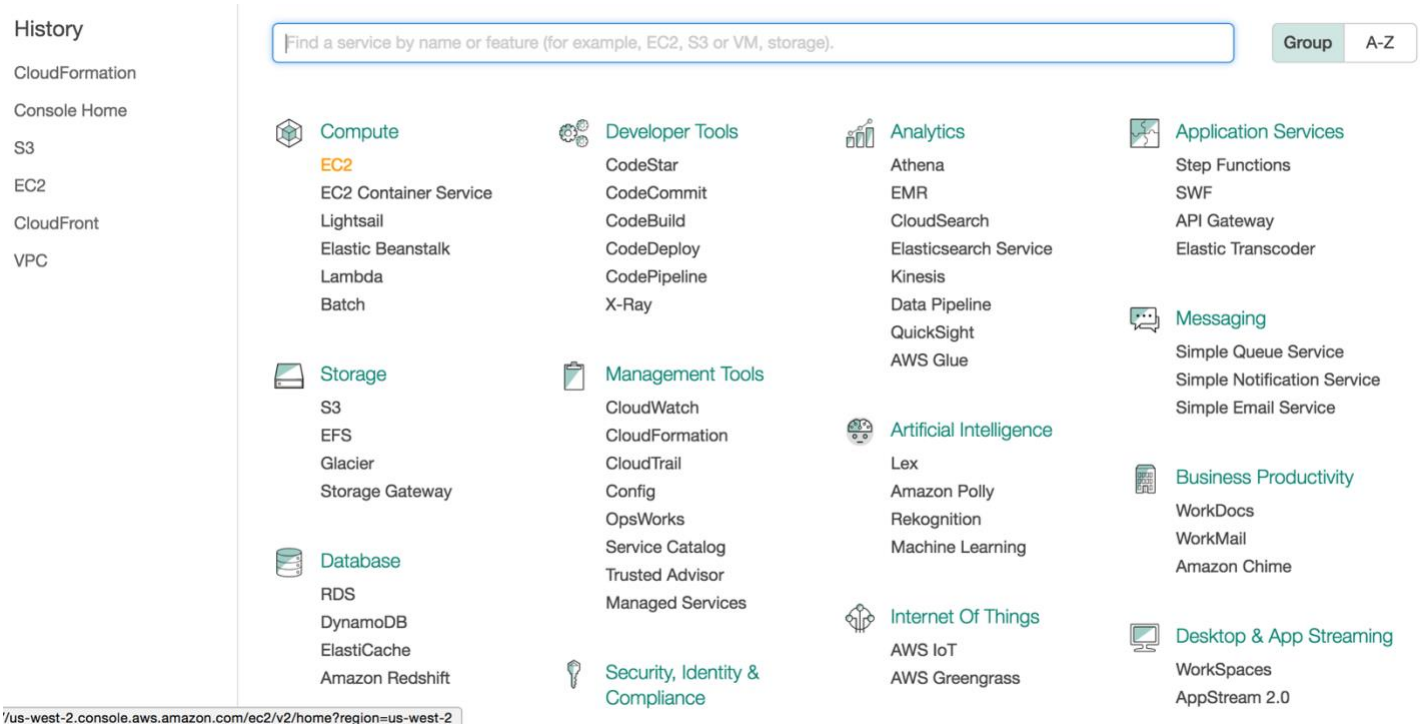
When the stack has been created you will see a **CREATE_COMPLETE** status. The automation is ready for use at this point.



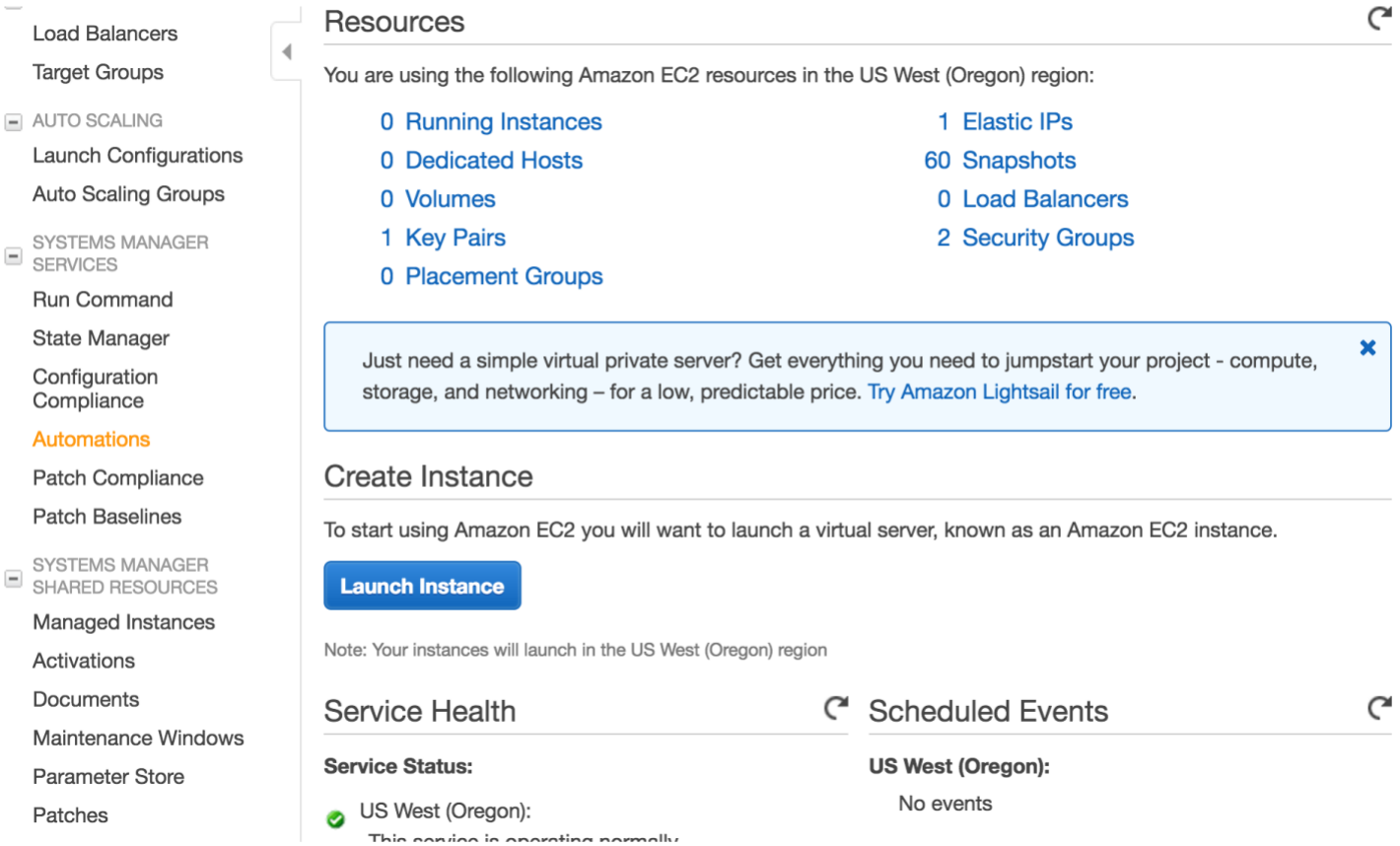
Running the Automation

To run the automation:

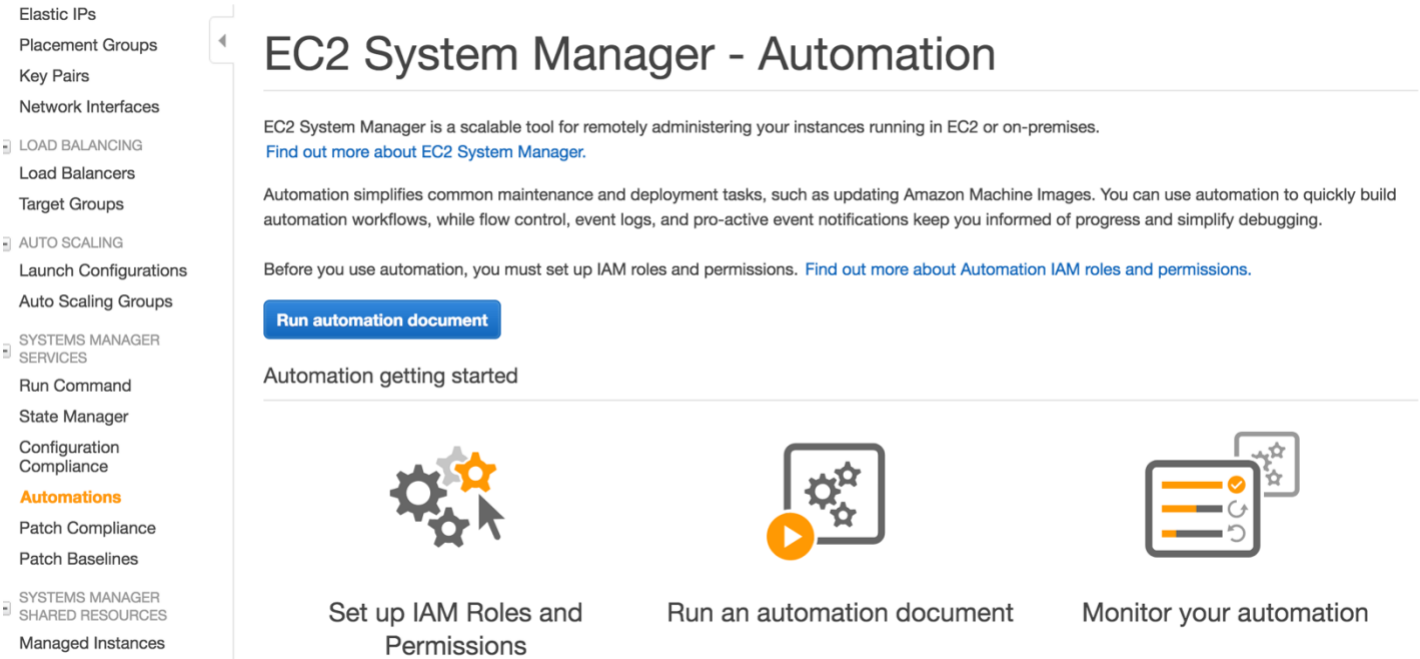
1. Log into your AWS console and go to EC2.



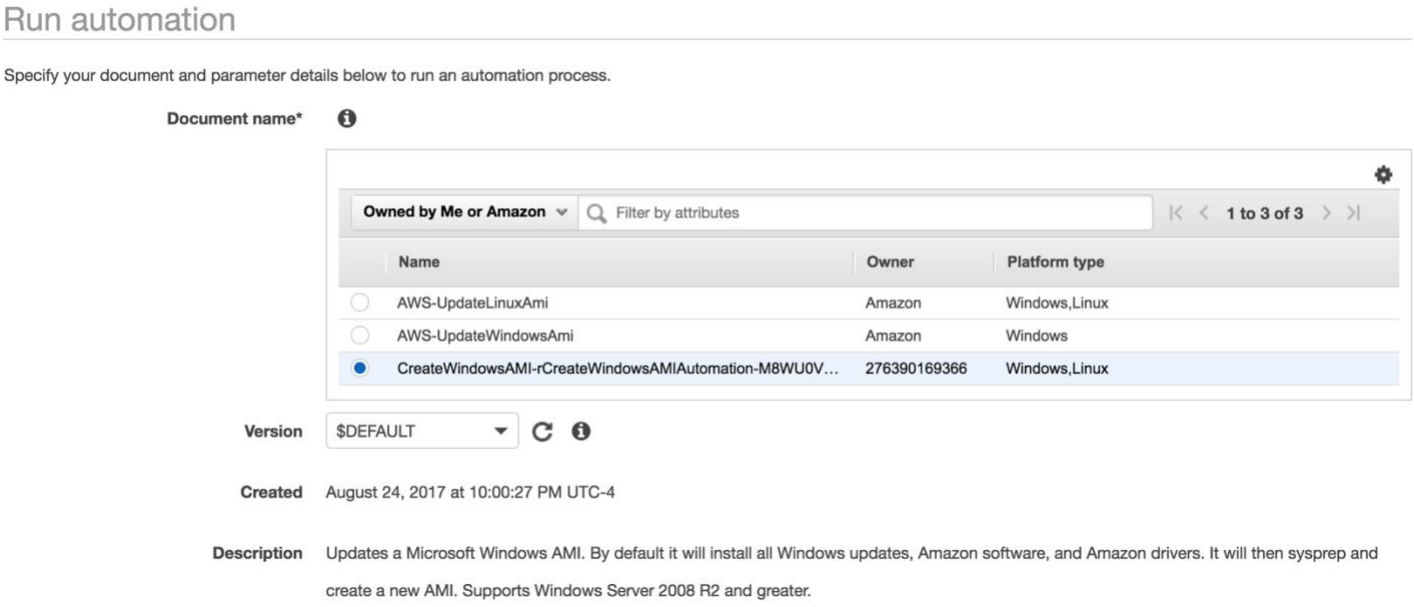
2. Under **SYSTEMS MANAGER SERVICES** in the left pane, click **Automations**.



3. On the **EC2 Systems Manager - Automation** page, click **Run automation document**.

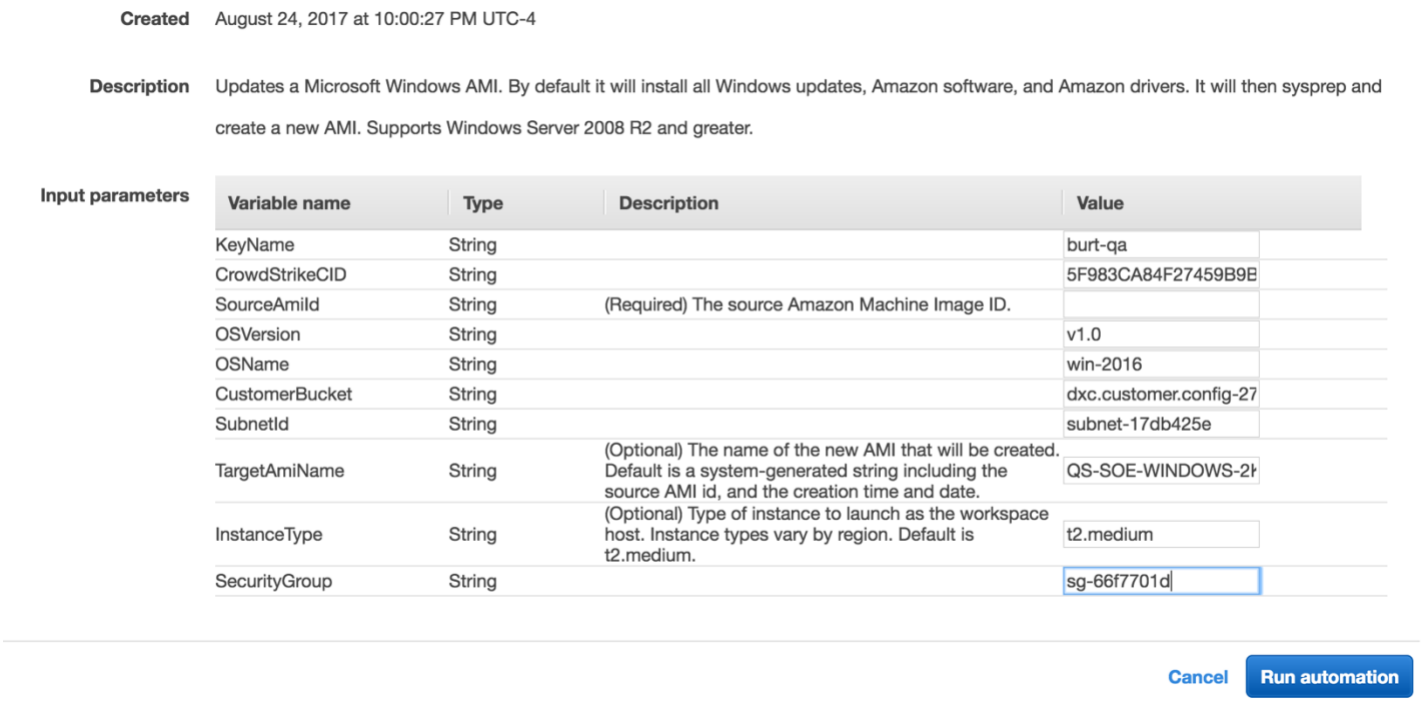


4. From the list of automation documents, select the Windows AMI creation automation that you created previously.



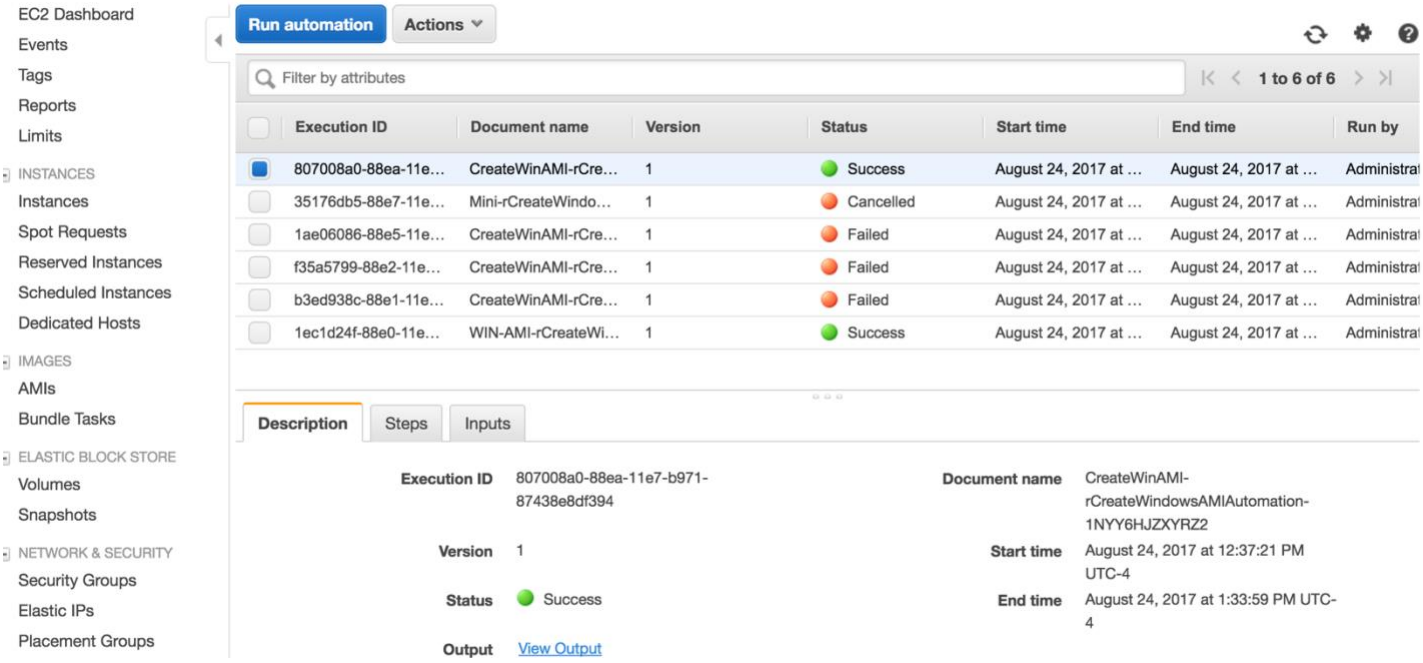
5. In the **Input parameters** section, configure the following parameters:

Parameter	Description
KeyName	A key in the region and account where the automation is being run.
CrowdStrikeCID	The customer identifier for Crowdstrike.
SourceAmiId	The AMI under which the new AMI will be created. This is our starting product. The default shown is the supported AMI we use from AWS for Windows 2016. If you wish to create a Windows 2012 R2 AMI (the only other version supported by our product), do a search in Quick Start list of AWS AMIs for "2012" and use the AMI ID for "Microsoft Windows Server 2012 R2 Base" for this field.
OSVersion	OS version. This needs to be unique under all OSName type AMIs for your region.
OSName	The name of the Windows version for which you are creating the AMI. Must be either "win2016" or "win2012".
CustomerBucket	The name of the customer bucket from which assets will be pulled.
SubnetId	The subnet to which the Source AMI will be connected.
TargetAmiName	This is the prefix used in the creation of the AMI name for the newly created Windows AMI.
InstanceType	The EC2 Instance Type used to launch the Source AMI-created virtual machine.
SecurityGroup	The security group on the VPC associated with the SubnetId to which the instance built upon the Source AMI is connected. This security group should allow for RDP (port 3389) access.

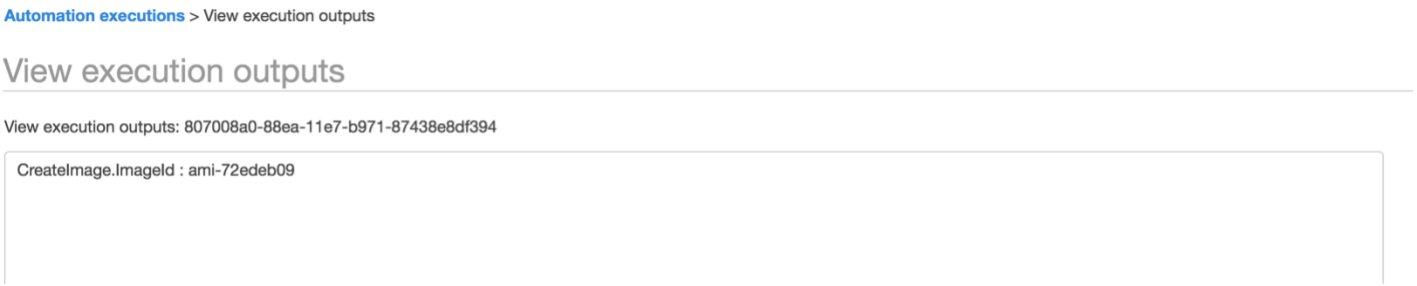


6. Click **Run automation**.

After the automation has successfully run, click the **View Output** link on the **Description** tab.



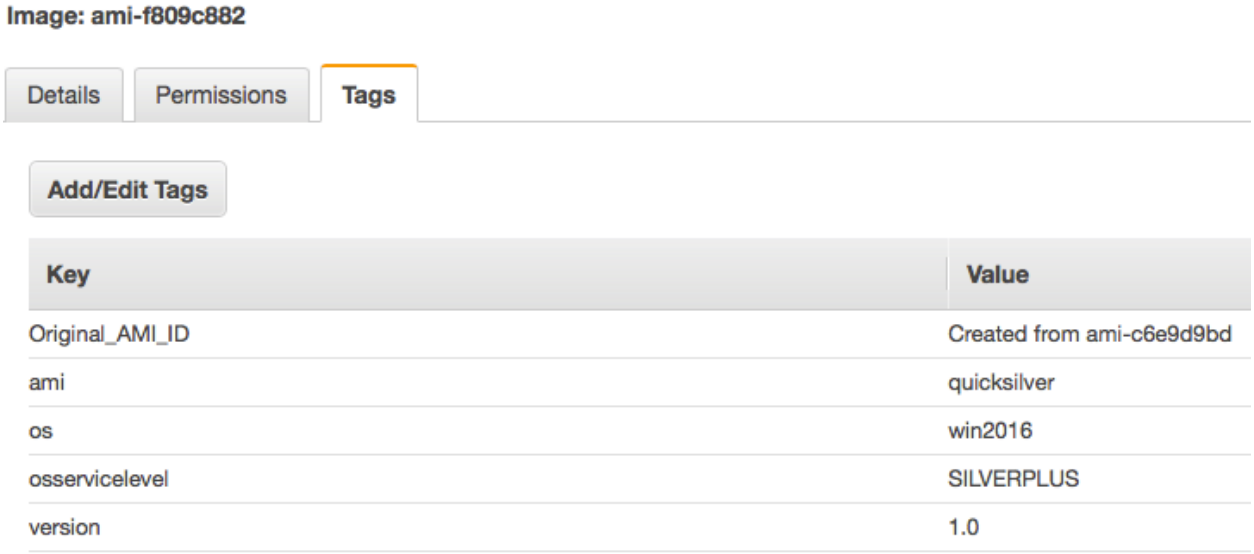
The id of the newly created AMI will be listed under **View execution outputs** from the **Actions** menu.



AMI Tags

The following tags should be applied to the AMI after it is created:

Key	Value
Original_AMI_ID	Created from <SourceAmiId value passed during ami creation>
ami	quicksilver
os	win2012 or win2016
osservicelevel	SILVERPLUS
version	<OSVersion value passed during ami creation>

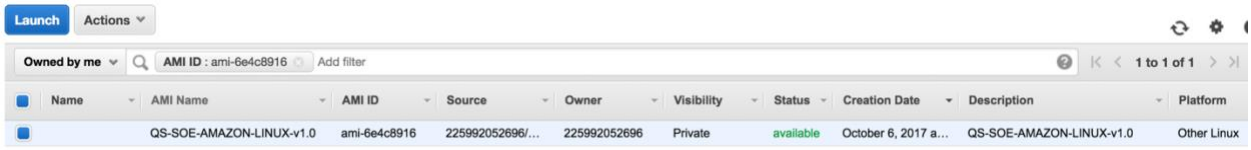


Creating an Encrypted AMI

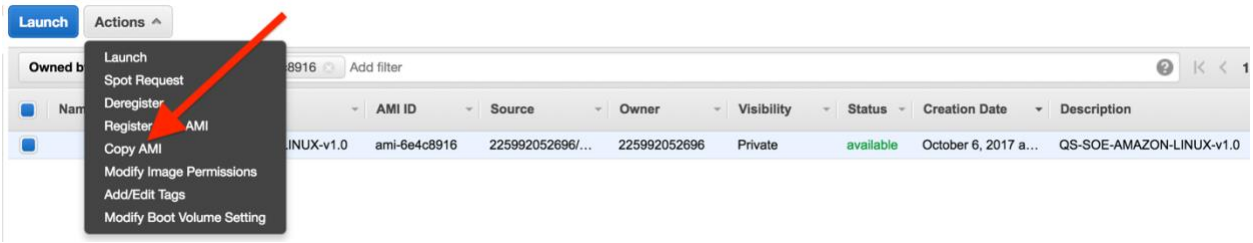
To create an encrypted AMI for either Linux or Windows, create the AMI as described in the *Creating a Linux AWS AMI* or *Creating a Windows AWS AMI* sections, and then encrypt the resulting AMI using the following procedure.

To create an encrypted AMI:

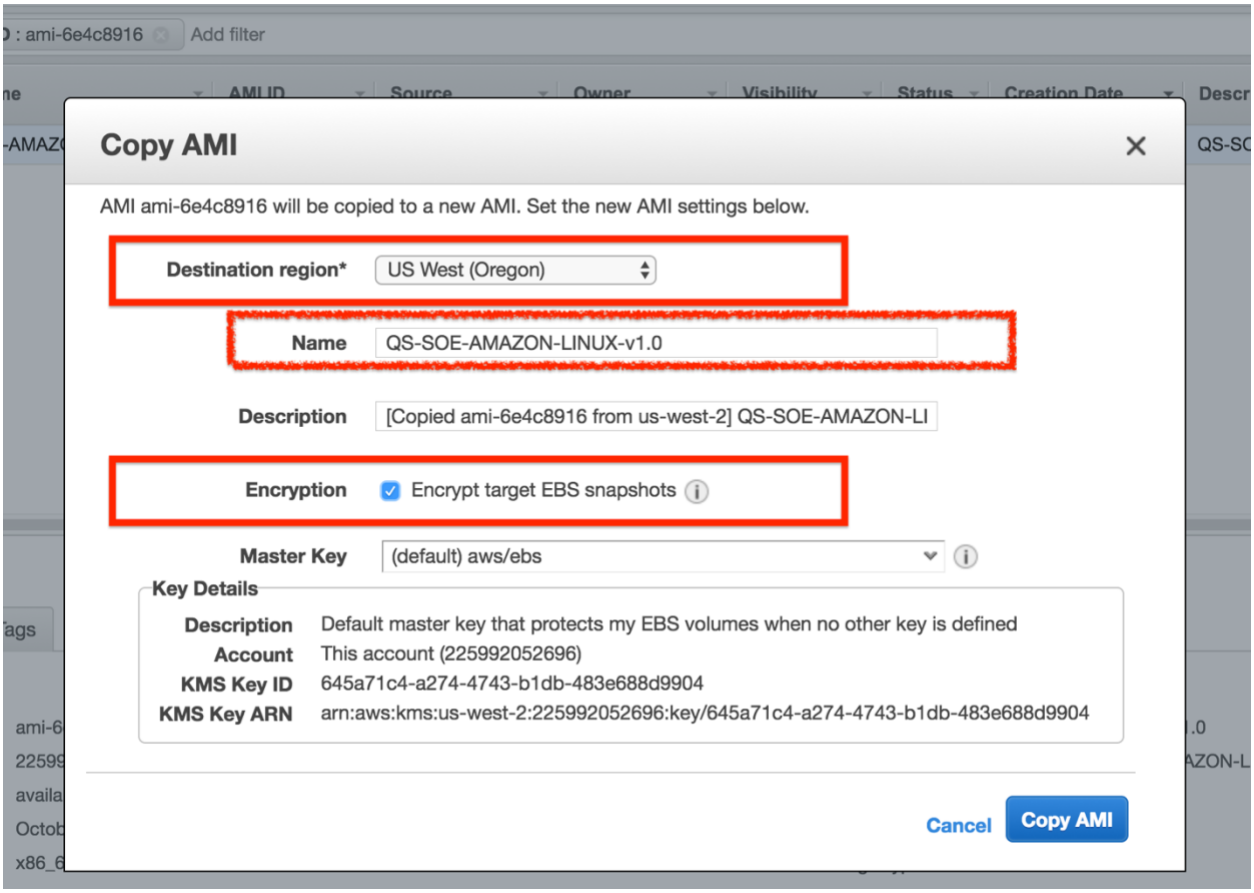
1. Log into your AWS console and go to EC2.
2. Under **IMAGES** on the left panel, click **AMIs**.
3. Select the AMI that was created using the automation document.



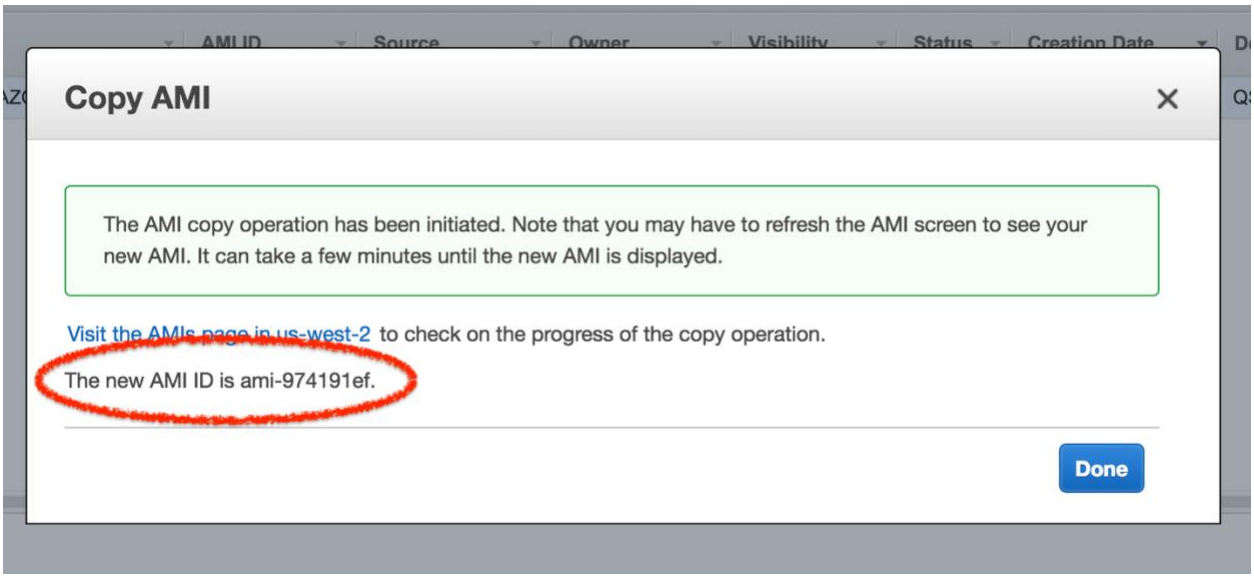
4. Click **Action** and select **Copy AMI** from the drop-down menu.



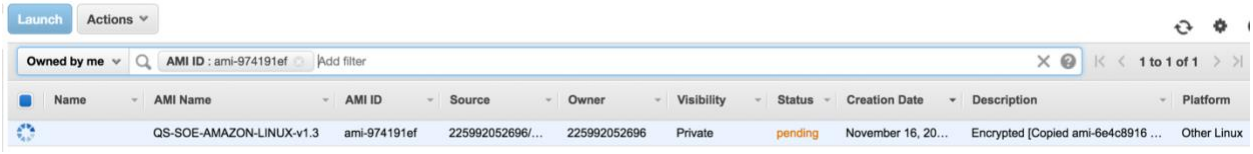
5. From the modal window change the **Destination Region**, **Name**, **Description** (optional) and click on the **Encryption** check box to enable encryption.



6. [Optional] Change the default Master Key if the customer is using a key with KMS.
7. Click **Copy AMI**. A notification window will be shown with the new encrypted AMI ID.



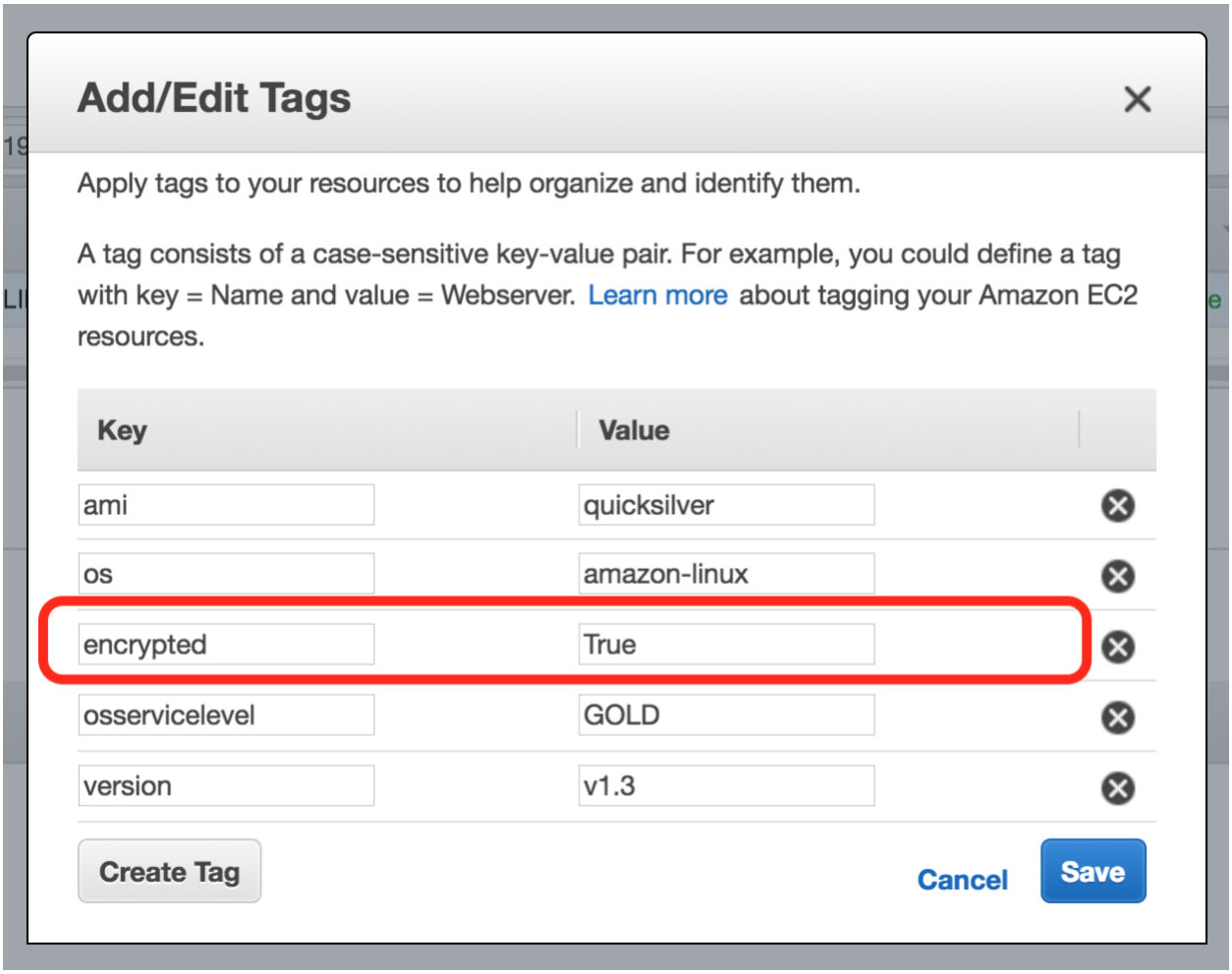
8. Initially, the AMI status will be pending and may take few minutes to become available.



AWS does not copy tags and permissions.

AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI.

9. Therefore, the new AMI needs to tagged appropriately. Click the **Add/Edit Tags** button and create the same tags that exist on the source AMI, but assign the **encrypted** key's value to **true** or **True** (case-insensitive).



10. Click **Save**.

9 Monitoring Infrastructure

The following CloudWatch alarms are defined when a new account is configured:

Alarm	Description
IAM Policy Changes Alarm	Any IAM policy change is made.
Security Group Changes Alarm	Any security group is created, changed, or deleted.
Unauthorized Attempt Alarm	More than 5 error responses of type AccessDenied and/or UnauthorizedOperation are returned by AWS in a 5 minute period.
CloudTrail Configuration Change Alarm	Any change is made to the CloudTrail Configuration.
Create Access Key Alarm	Any AWS Access Key is created.
Root Activity Alarm	Any action is taken by the root user, other than service events.
Network ACL Changes Alarm	Any VPC Network ACL is added, changed, or deleted.

An alarm will initially appear in the INSUFFICIENT_DATA state until the first event for that alarm occur.

Configuring Gold Managed Services

10

Gold Managed Services use native AWS services except for antivirus, which uses a SaaS-based solution from CrowdStrike. All the Lambda functions required to support the services are created automatically from the Master template. With the exception of Patching for Linux, all the Gold Managed Services are configured automatically from the Master template.

Gold Managed Services include the following:

1. **Automated Scheduled VM Backups**
The automated backup service uses the Elastic Block Store (EBS) Snapshot feature to create point-in-time snapshots of an EBS volume. A scheduled job runs every day to create a backup of the volume. By default, snapshots of EBS volumes are stored for 30 days. Snapshots are incremental backups, which means that only blocks changed on the device since the last snapshot are saved. When you delete a snapshot, only data unique to that snapshot is removed. Active snapshots contain information needed to restore your data from the time the snapshot was taken to a new EBS volume. For more information about AWS EBS snapshots, navigate to http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.

2. **Patching**
The automated patch management service uses AWS Systems Manager for Windows and a DXC-native solution leveraging EC2 Run Command for Red Hat Enterprise Linux patch management. The Patching section provides details on how to use each service.

3. **Monitoring VMs 24x7**
Amazon CloudWatch is a monitoring tool used to monitor AWS resources and applications on AWS. AWS CloudWatch continuously monitors the VM track performance metrics and triggers alarms at thresholds provided out of the box (see monitoring section) or based on custom thresholds. When an alarm is triggered, an incident is created and inserted into the ServiceNow CMDB for a Support Engineer's response and remediation. For more information about CloudWatch, navigate to https://aws.amazon.com/cloudwatch/

4. **Endpoint Protection**
The Endpoint Protection solution uses CrowdStrike Falconhost, a cloud-based antivirus service where an agent (Falcon Host sensor) is deployed in the managed instance. Installation is verified via the Falcon Host management interface. The CrowdStrike Falcon host system is integrated with the DXC Security Operations Center (SOC)

5. **Remote Instances Management**
Instances are managed using the EC2 Systems Manager Run Command feature. Run Command enables you to automate common administrative tasks and perform ad hoc configuration changes at scale. If for some reason Run Command is failing to accomplish an administrative task on a single instance or collection of instances, a remote bastion/jump box service has been provided. This service provisions an ephemeral bastion box for Windows or a jump box for Linux in the Management VPC. All actions taken from either the Run Command or Bastion/Jump Box service are logged using CloudTrail. It is highly recommended that you perform all administrative tasks using Run Command and that you use Bastion/Jump Box only as a fail-safe option. The Remote Instances Management section provides details on how to use these features.

Performing Backup Service and Health Checks

The backup service relies on the AWS EBS Snapshots mechanism. After an instance is tagged for backup, the backup service takes a snapshot of the EBS volumes on that instance according to a predefined schedule. The backup service also cleans up the old snapshots from AWS using the retention

time specified on the instance. The backup service is created automatically from the Master template. The following section describes the backup service..

Components

- Instances to be backed up are tagged during creation time. Three tags are created on the instance based on user prompts:
  - **Backup** = true/false - Configures whether to back up the volumes or not. This value is copied from the subnet since it cannot be chosen per workload.
  - **BackupSchedule** - Leave blank to use the default backup schedule, or enter the name of a Custom Backup Schedule (see Creating Custom Backup Schedules).
  - **RetentionPeriod** - Controls the number of days a backup is maintained. The default is 30 days. The default can be overridden when deploying the master stack. That value (number of days) will be stored in a tag on the subnet. When creating a workload instance, the retention period can be set to the subnet's default, or overridden to a different number of days.
- A CloudWatch Event Rule that includes "**rSnapshotCreateRule**" in its name is created to trigger the backup function based on a Cron expression.
- A CloudWatch Event Rule that includes "**rSnapshotDeleteRule**" in its name is created to trigger the cleanup of snapshots that exceed the Retention Period.
  - Both events are scheduled at 15 minutes past midnight UTC and run concurrently. Modify this schedule by editing the CloudWatch event definitions.
- A CloudWatch Event Rule that includes "**rHealthScheduleRule**" in its name is created to trigger verification that backups are being created correctly.
  - The event is scheduled at 15 minutes past 1pm UTC. Modify this schedule by editing the CloudWatch event definition.
- A Lambda function "**backupHandler**" triggered by the CloudWatch Event Rules performs the backups and purges expired backups when executed.
- A Lambda function "**backupHealth**" triggered by the CloudWatch Event Rules verifies that backups are being created correctly when executed.
- Two CloudWatch log groups, **/aws/lambda/backupHandler** and **/aws/lambda/backupHealth,** are created to hold events from the backup process and health monitoring.
- An SSM automation document that includes "**CreateLinuxBackupSnapshot**" in its name creates the EBS snapshots for Linux workload instances. The **backupHandler** Lambda function executes this automation document for each volume attached to a Linux instance.
  - The automation document performs an "fsfreeze" command to quiesce the filesystem on the volume, then takes a snapshot of the volume by calling the EC2 create-snapshot command, then unfreezes the filesystem.
- An SSM automation document that includes "**CreateWindowsBackupSnapshot**" in its name creates the EBS snapshots for Windows workload instances. The **backupHandler** Lambda function executes this automation document for each volume attached to a Windows instance.
  - The automation document runs Windows PowerShell commands to call the AWS New-EC2Snapshot command, utilizing Windows Volume Shadow Copy Service (VSS) to safely take a backup of the volume, even if applications are writing to it.

Rules

All attached volumes on an instance are backed up, including the root volume. Tags are applied to the snapshots using these rules:

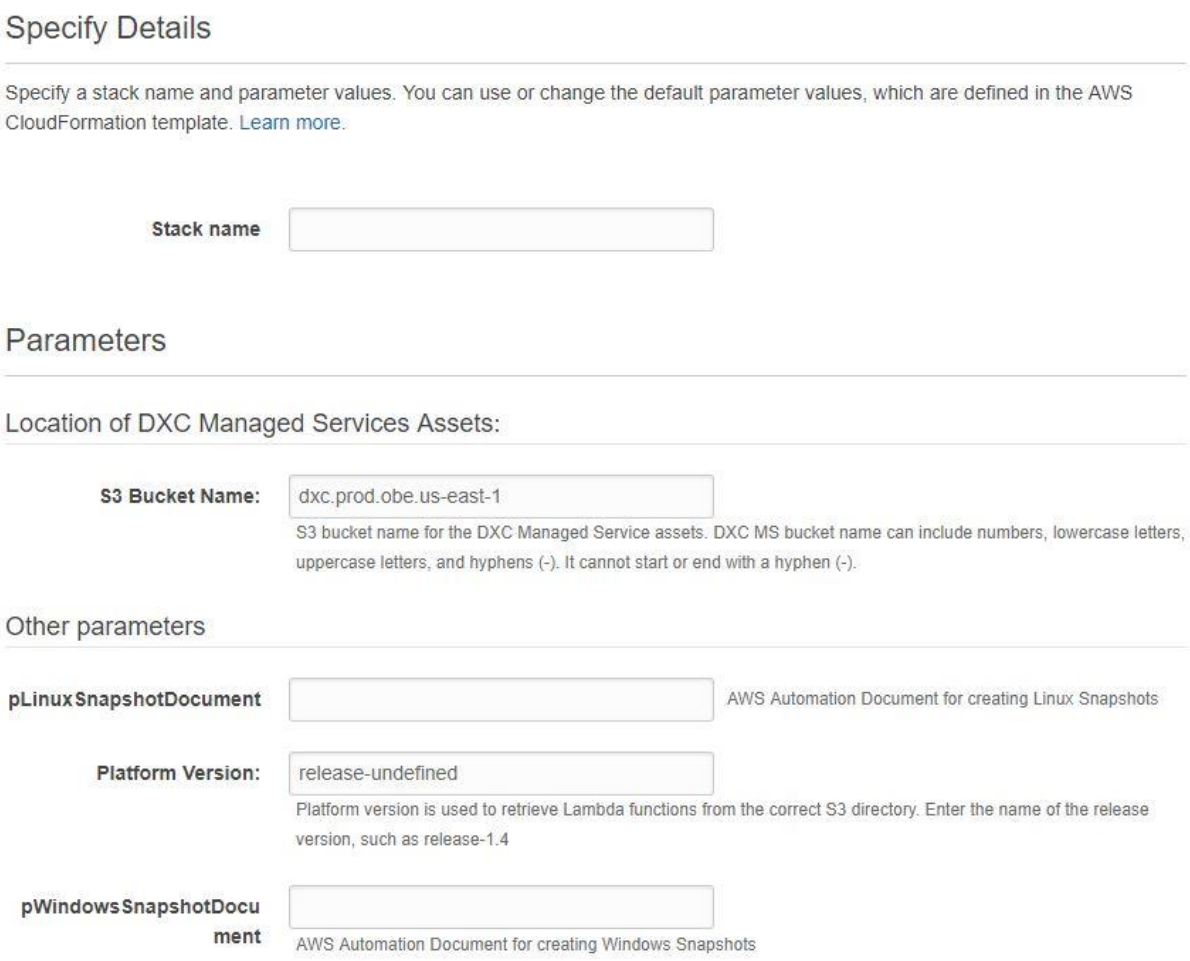
- Tags from the volume are copied to the snapshot.
- Additional tags are then added:
  - DeleteOn: Represents the date the snapshot should be deleted. The date is calculated using the current date and the Retention Period. The format is YYYY-MM-DD.

- Name: String InstanceId: <EC2 Instance ID>
- VolumeName: The InstanceName tag from the instance tags with the device name appended; for example, MyWebServer-/dev/xvda.

To install the backup components:

1. Run the **SnapshotLinuxVolumeAutomation.json** CloudFormation template to create theEC2 SSM automation document that is executed when a backup is run for Linux workload instances.

2. On the **Specify Details** page, enter a **Stack name** (for example, *Backup-Linux-Automation*).
3. Run the **SnapshotWindowsVolumeAutomation.json** CloudFormation template to create the EC2 SSM automation document that is executed when a backup is run for Windows workload instances.
4. On the **Specify Details** page, enter a **Stack name** (for example, *Backup-Windows-Automation*).
5. Run the **Backups.json** CloudFormation template to create the backup components.



6. On the **Specify Details** page, enter a **Stack name** (for example, *Backup-Service*).
7. Enter the name of the S3 bucket that contains the Managed Services assets.
   The S3 Bucket has a region-specific suffix, such as ".us-east-2"
8. For the **pLinuxSnapshotDocument** field, enter the name of the Linux snapshot document created when you executed the **SnapshotLinuxVolumeAutomation.json** template. To locate this document name, select the stack you created from the list of stacks in the CloudFormation list with the name you gave it (for example, *Backup-Linux-Automation*). Click the **Resources** tab and copy the Physical ID of the **rSnapshotLinuxVolumeAMIAutomation** resource.
9. For the **pWindowsSnapshotDocument** field, enter the name of the Windows snapshot document created when you executed the **SnapshotWindowsVolumeAutomation.json** template. To locate this document name, select the stack you created from the list of stacks in the CloudFormation list with the name you gave it (for example, *Backup-Windows-Automation*). Click the **Resources** tab and copy the Physical ID of the **rSnapshotWindowsVolumeAMIAutomation**resource.
10. Enter the Platform Version for the current version of the product, which is "release-1.4". Do not leave this value defaulted to "release-undefined".
11. Click **Next** through the remaining screens to create the template.

The Backup AWS Lambda function handles the following:

- Filters the instances based on the **Backup** tag and performs the following:
- Creates a snapshot of each attached volume.
- Calculates the snapshot's Delete-On date using the Retention tag value and then tags the value on the snapshot as **DeleteOn**.
- Tags the snapshot with additional tags as described above.
- Deletes snapshots that exceed the Retention Period.

The CloudWatch logs for the account show the steps taken, snapshots created, and snapshots removed.

Snapshot Health Check

The Snapshot Health Service performs the following:

- Verifies that a snapshot with a recent date is present for all volumes attached to instances tagged for backup.
- Detects when instances are added to a non-existent backup schedule.
- Generates CloudWatch logs when anomalies are detected in the backups.
- Generates email alerts to the Account Email (configured during logging setup) when anomalies are detected in the backups.
- Generates an Event to ServiceNow when anomalies are detected in the backups.

To determine backup health, examine the recent snapshot for a volume and compare it with the time of the last triggered backup schedule for the instance. If the instance was created after the last scheduled backup or the volume was created and attached after the last backup, no health can be determined. If the backup is older than the last scheduled backup, an alarm is raised and a notification is issued so that the snapshots can be investigated by a Delivery Engineer.

The Backup Health Service runs 1 hour after the Default backup schedule each day. If the schedule for the Default backups is modified, the schedule for the Backup Health Service should also be modified. The health checks continue to work regardless of the Default backup schedule and the Health schedule.

When Custom backup schedules are used, the health checks verify that the most recent backup was completed successfully based on the custom schedule. For example: A custom schedule defines backups to run at 10 a.m., 2 p.m., and 8 p.m. The backup health checks are scheduled to run at 12 p.m. A snapshot should exist that is approximately 2 hours old when the health checks are performed. If this snapshot is not present, an alarm is raised and the most recent backup is reported. It takes several minutes or more to create a snapshot; therefore, backups initiated at 10 a.m. might be tagged with a create time stamp slightly past 10 a.m. The backup health check allows a variance of 1 hour when determining the difference between the backup schedule and the actual time of the snapshot.

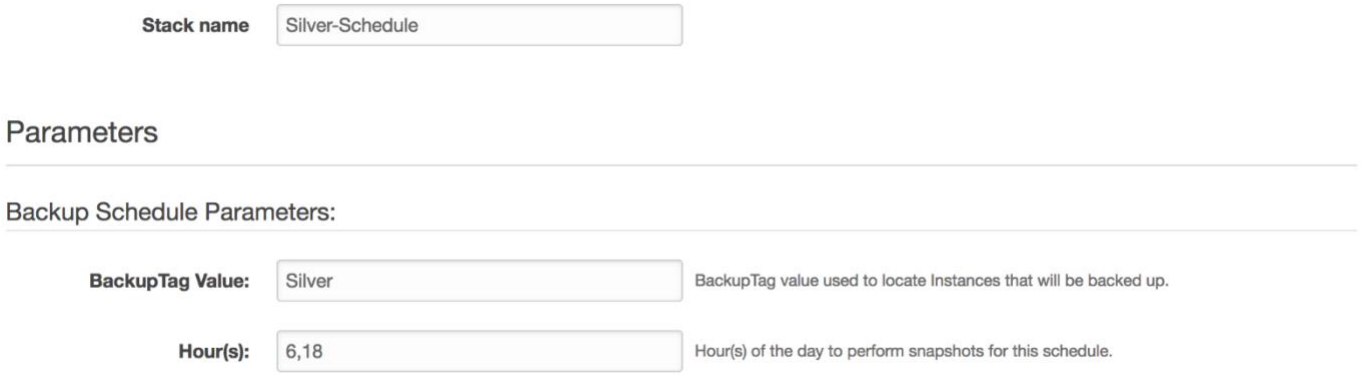
Creating Custom Backup Schedules

By default, snapshots are created once a day for volumes of backup-enabled instances. Some applications might require that volumes are saved more frequently. To support more frequent backups, you can create custom backup schedules where a backup happens multiple times per day.

Creating a Custom Schedule

Create a custom schedule using the *CustomBackupSchedule.yaml* CloudFormation template. This template is copied to the Customer bucket during account setup or upgrades.

1. Select the *CustomBackupSchedule.yaml* template in CloudFormation.
2. Specify a unique name for the **Stack name**.



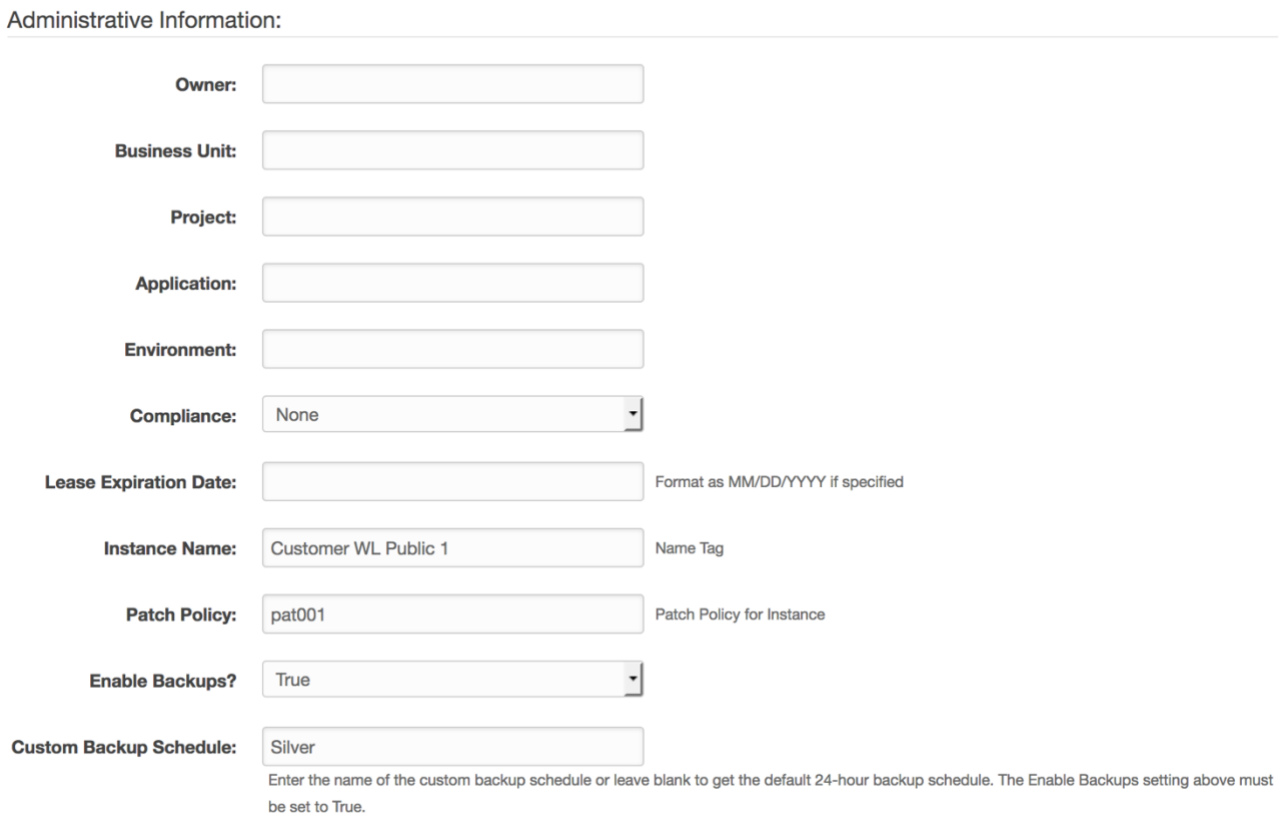
3. Specify a **BackupTag Value**. This value is used to tag instances that will be backed up using this schedule.

The hours of the day (UTC) are specified as a list of hours from 0 to 23. Zero represents midnight UTC. By offsetting the hour values by the local time zone, you can schedule backups to take place at appropriate times for your instances. In the example, the backups would occur at midnight CST (UTC-6) (the hour value of 6) and noon CST (the hour value of 18).

Executing the template results in a CloudWatch event that triggers the backup processing to occur at the specified hours. All instances with a backup schedule of Silver will be backed up at these times.

Provisioning Instances

When you provision instances using the CloudFormation templates, you can specify a backup schedule. The **Administrative Information** section of the provisioning template contains options for the backup schedule:

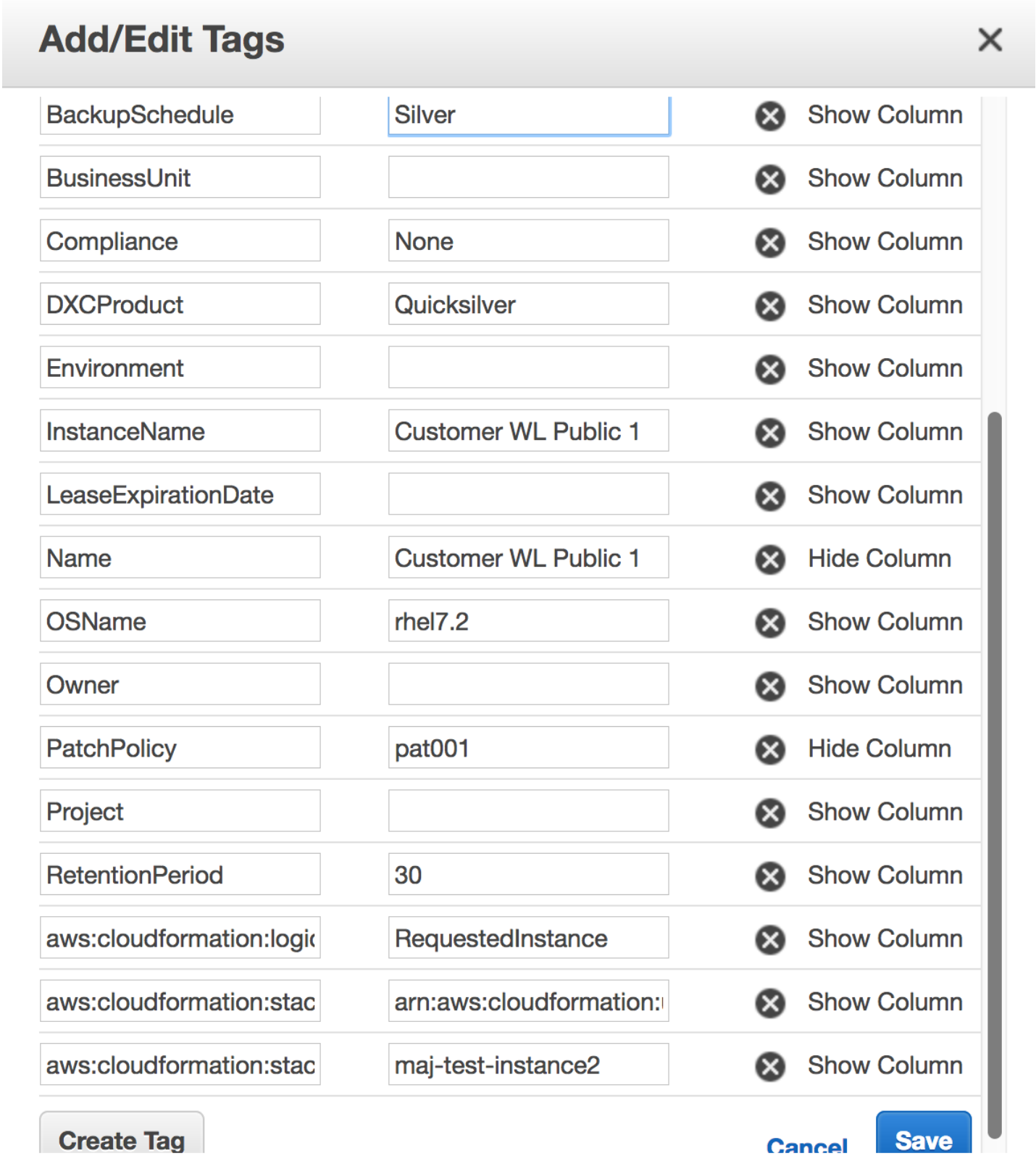


1. Set the **Enable Backups** option to **True** (the default).
2. Enter the name of the **Custom Backup Schedule** (for example, Silver).
   Entering an invalid schedule name results in an error message during backups and backup health checks and the instance's volumes are backed up during the standard daily backups.

Applying a Custom Schedule to Existing Instances

1. If you create a custom schedule and you want to apply it to existing instances, follow these steps:
2. In the AWS Console, browse to the **EC2 Service**.
3. List the running instances.
4. Select the desired instance.
5. On the bottom of the screen, click the **Tags** tab.
6. Click **Add/Edit Tags**.
7. If the BackupSchedule tag already exists, enter the name of the desired backup schedule. If the BackupSchedule tag does not exist, click **Create Tag**:
8. Enter **BackupSchedule** for the name of the tag.
9. Enter the name of the desired backup schedule.
10. Click **Save**.

Example tags:



The following rules determine which backup schedule is used for an instance:

1. The tag *BackupSchedule* determines the name of the schedule to use for the instance.

2. If the name of the schedule refers to a non-existent schedule, the normal backup schedule is used (once per day).
3. If the BackupSchedule tag is blank, the normal backup schedule is used (once per day).
4. If the BackupSchedule tag contains the value *Default*, the normal backup schedule is used (once per day)
5. If the BackupSchedule tag is not present, the normal backup schedule is used (once per day)

Using Backup Health Checks

Using a Custom Backup Schedule, the backup health checks check for a recent backup based on the custom schedule.

The logic in the backup health check examines the backup schedule and finds the most recent backup hour compared to the current hour. If the custom backup schedule specified the hours of 0, 6, and 18 and the health check ran at hour 10, then the health check would look for a backup done at hour 6. If this backup was not present, an alarm message would be generated and would show the most recent backup time (or none, if no backups had occurred). This allows the schedule for the backup health checks to be varied throughout the day while still making sure a valid backup exists.

Patching Windows and Linux Instances

11

AWS supports a native patching solution called Patch Manager for Windows and Linux instances using EC2 Systems Manager.

Patching Windows Instances

Patching of Windows instances is done via Patch Manager from AWS, which is a sub-service of EC2. The service allows users to manage patch baselines and maintain patch compliance.

This example illustrates how to create a patch baseline for a patch group of Windows instances. The baseline is applied under a maintenance window. This maintenance window sets the schedule for applying the patches (baseline) on the specified instances in the patch group. The patch group is set on the instance by setting the Patch Group tag.

Patch Baseline

The patch baseline is used to specify the following criteria about the patching to apply to instances:

- Classification (patch type)
- Severity
- Auto approve delay
- Approved (whitelisted) patches
- Rejected (blacklisted) patches

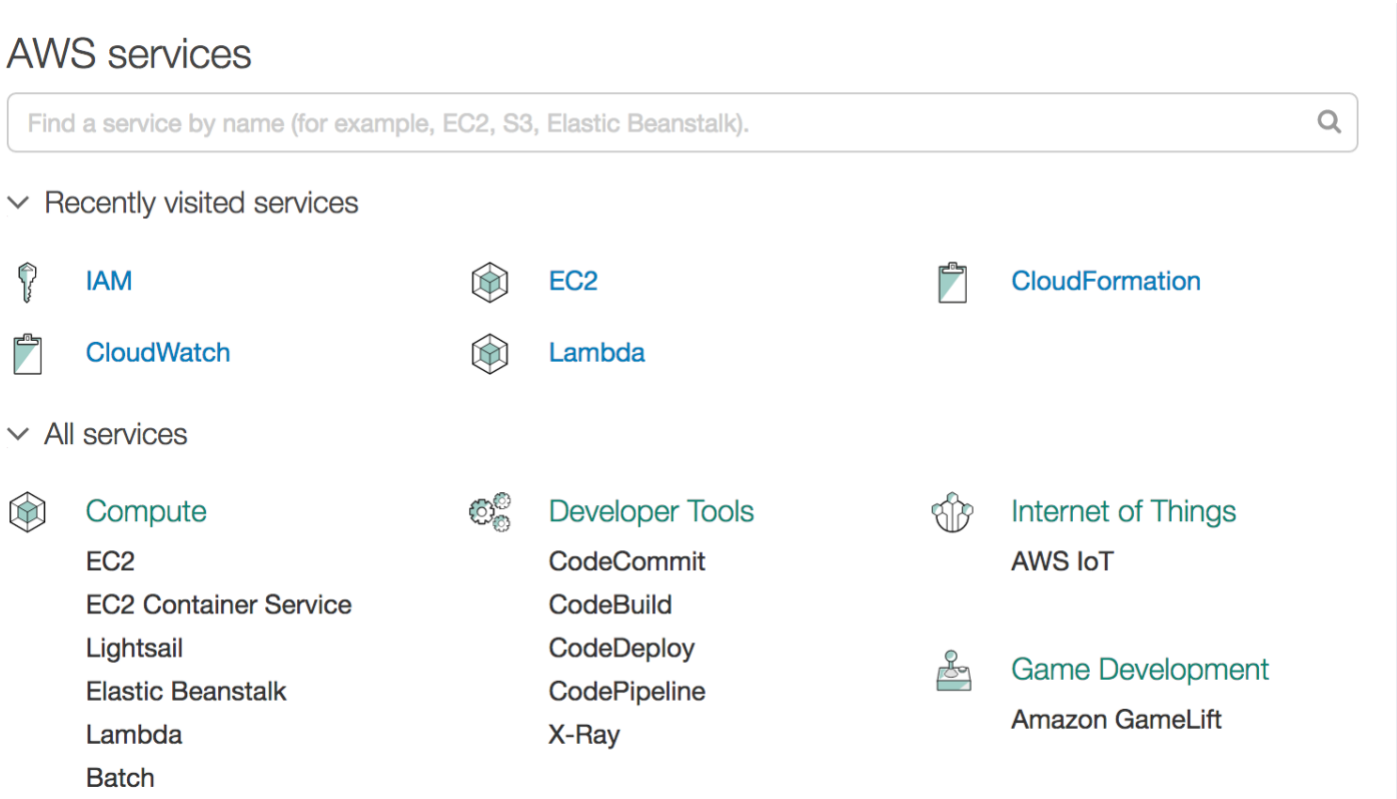
You can search for an existing patch baseline that meets your needs or create a patch baseline to apply to your instances.

Inspecting Current Patch Baselines

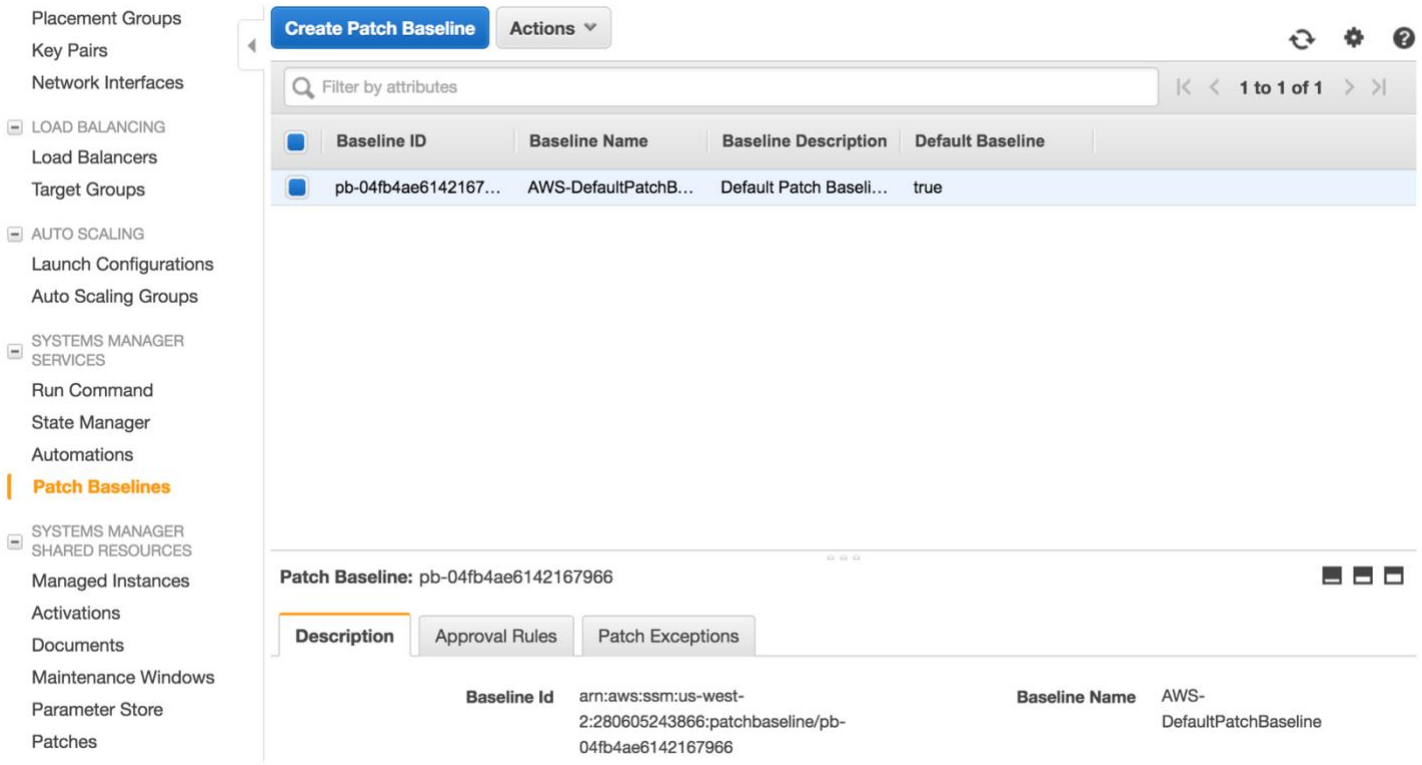
You can skip this section if you are creating a new Patch Baseline. For more information, see the "Create a Patch Baseline" section.

To inspect current Patch Baselines:

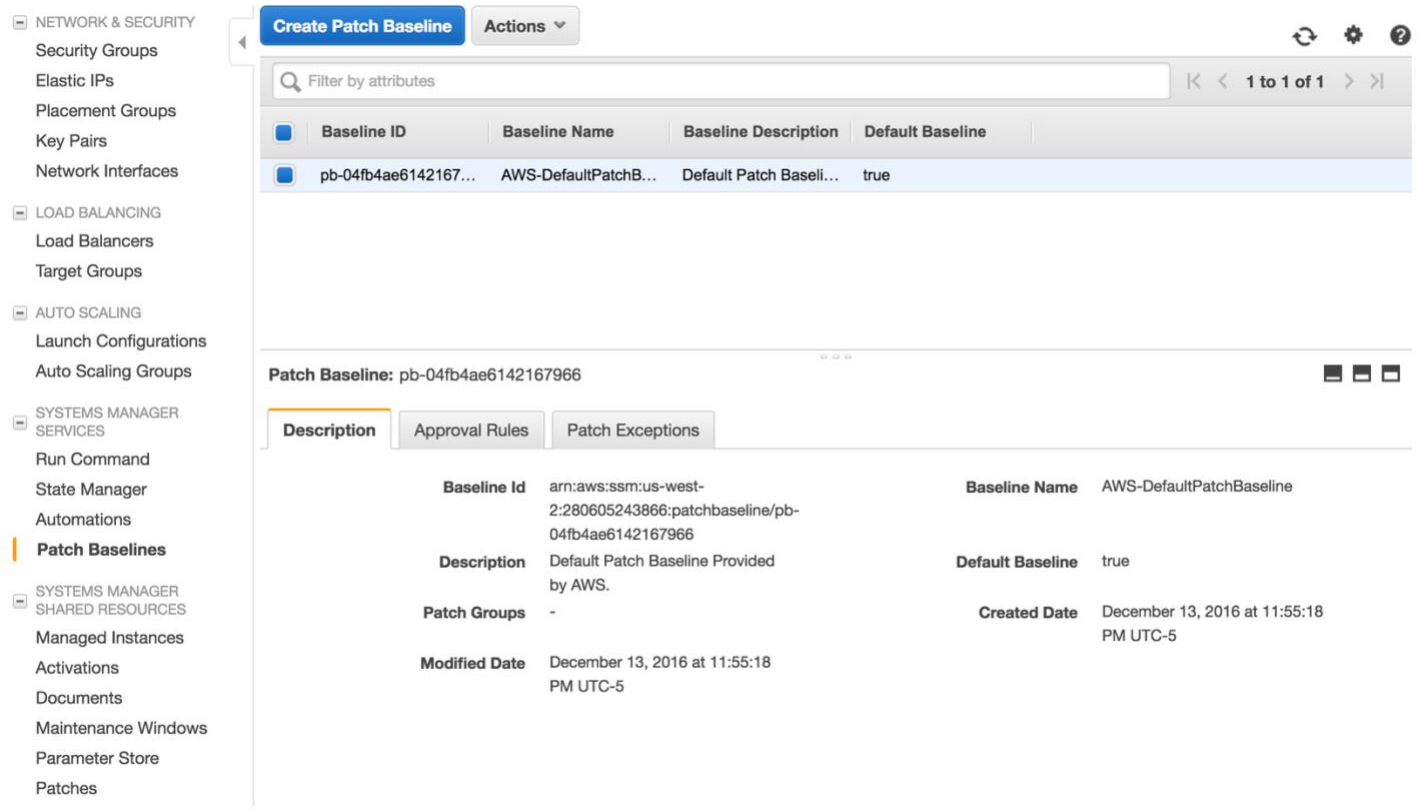
1. On the AWS services page, click **EC2**.



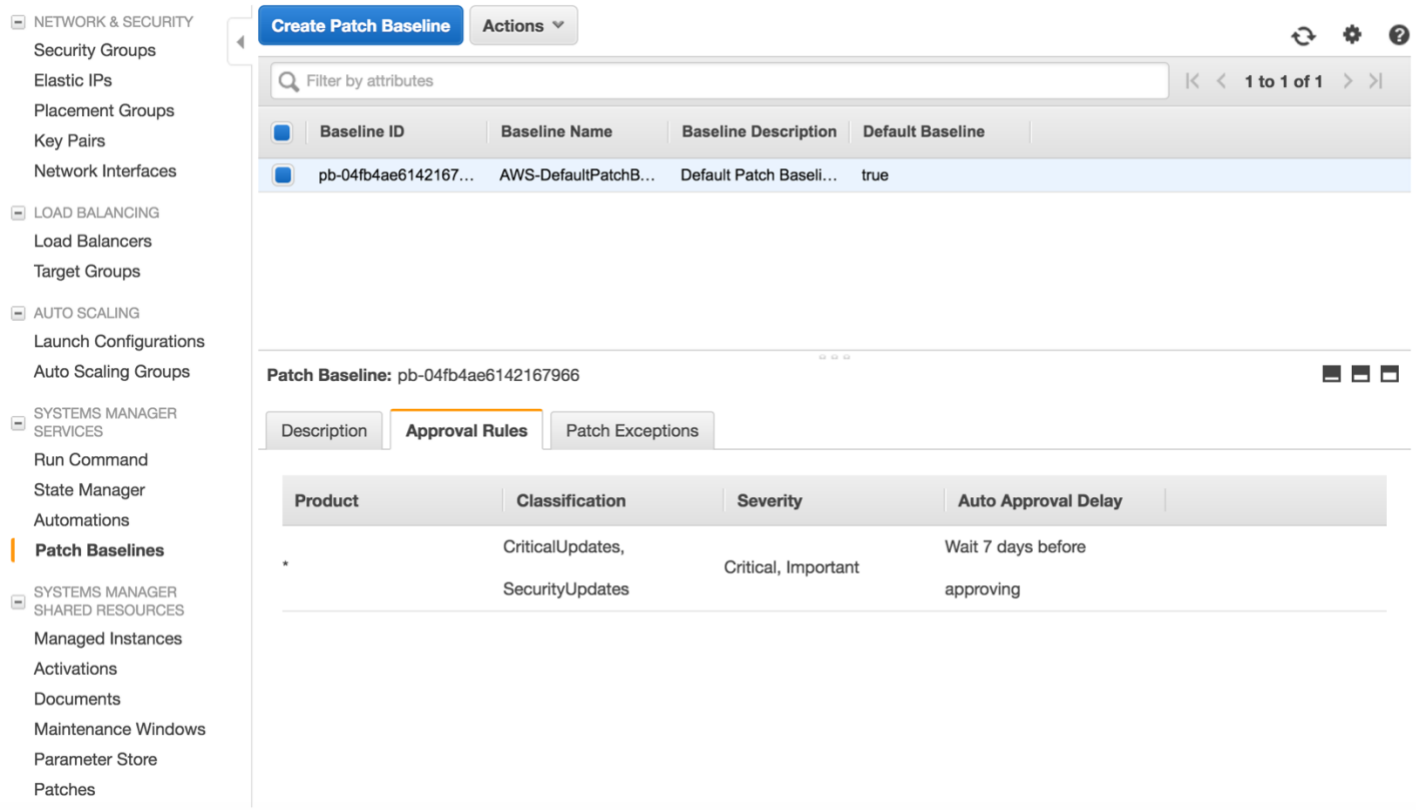
2. Under **SYSTEMS MANAGER SERVICES** on the left, click **Patch Baselines**.



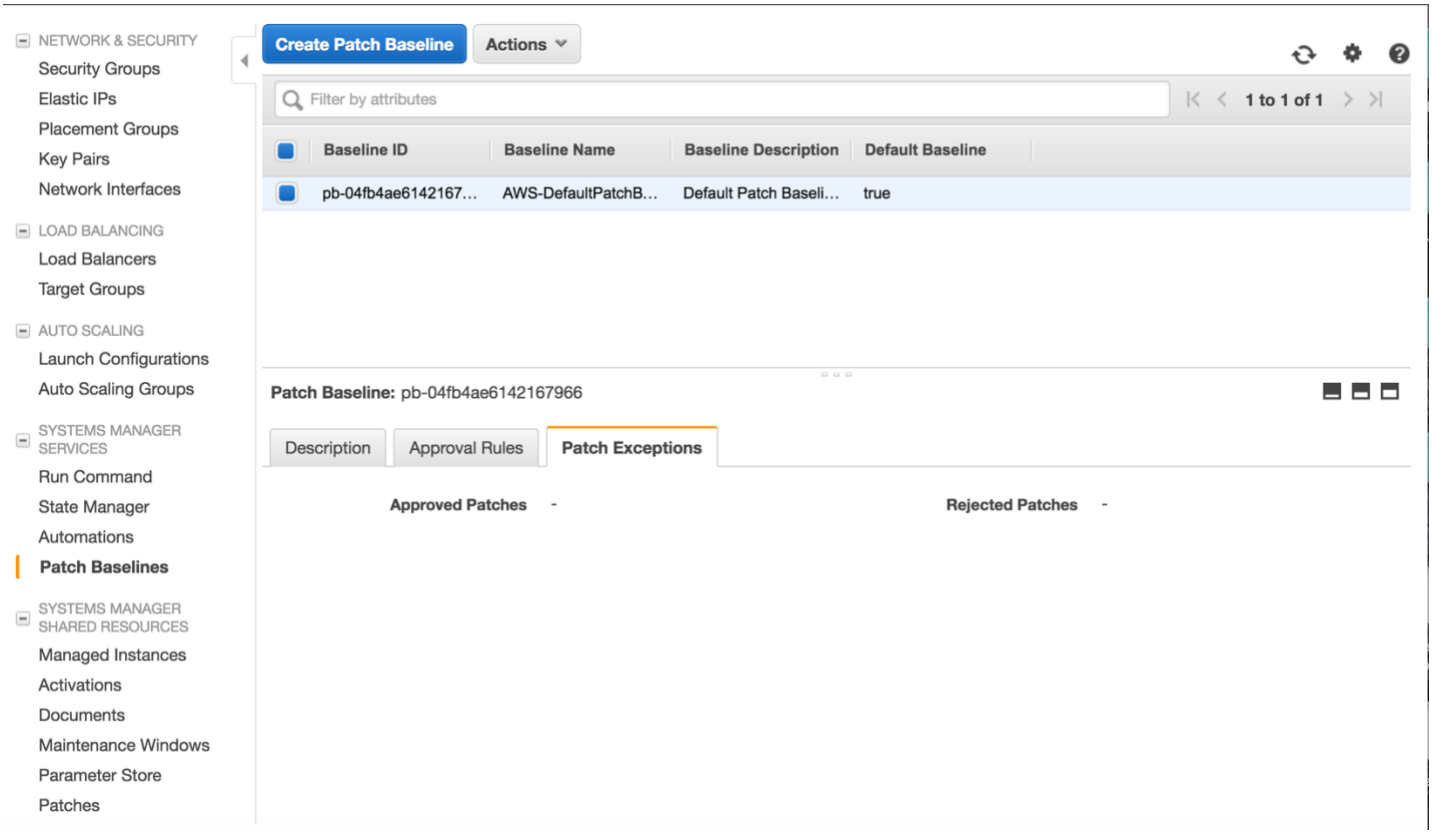
3. Select each patch baseline to view the description tab, approval rules tab, and patch exceptions (whitelist and blacklist).



The first example baseline applies critical and security updates of Critical or Important severity. The patches are applied 7 days after they are released.



This baseline does not have approved (whitelist) patches nor does it have rejected (blacklist) patches



Notes:

1. Because you can filter the list only by NAME_PREFIX and OWNER, it is essential that you name the patch baseline and its mapping to patch type for easy recognition of which baseline to apply.
2. Search for applicable patch baselines by executing the above procedure for each patch baseline until you find an existing one that works or you can create your own patch baseline.

Creating a Patch Baseline

Use a naming convention for patch baselines:

<application name>-<layer>-<environment>-<policy>-<region>

The example shows how to create a baseline for the production web servers for an application named *sampleapp* that runs under policy *pol001* in the *us-west-2* region.

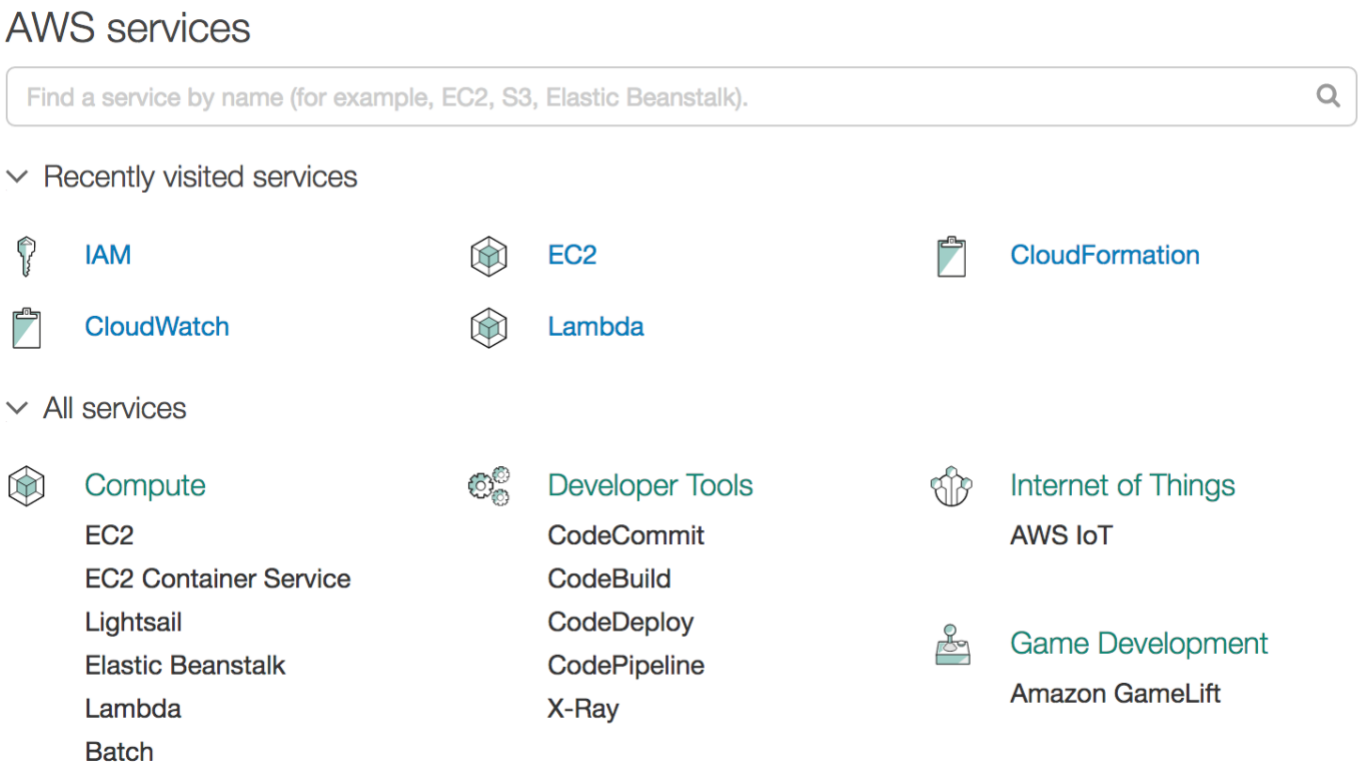
sampleapp-webservers-production-pol001-us-west-2

The patch baseline has the following requirements for the web servers, which are Windows 2012 R2 machines:

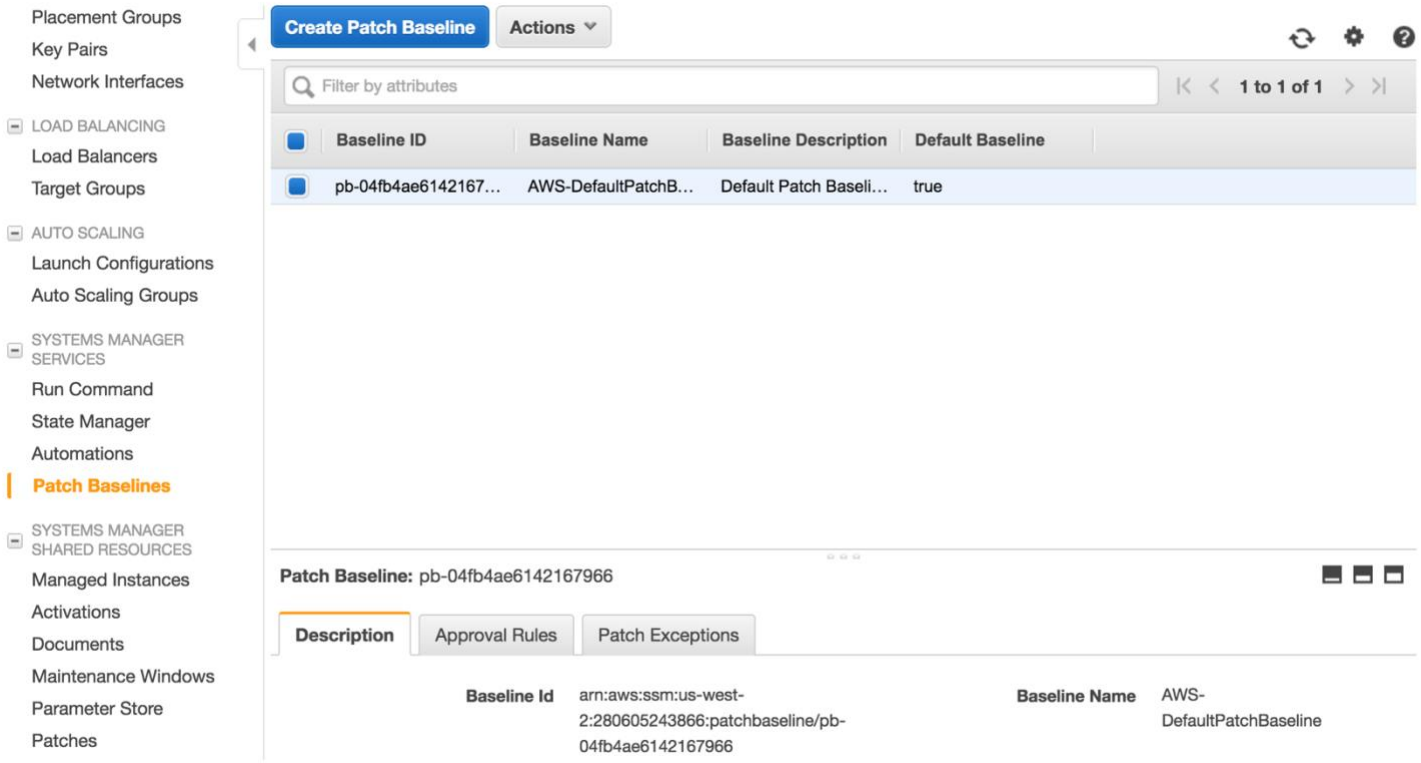
- Apply critical updates of severity Critical 7 days after release
- Apply security updates of level Important 5 days after release
- Do not apply patch KB4012216 (blacklist)
- Apply KB2919355, which is a security update of level less than Important
- Apply the standard non-security, non-critical update KB3210083

To create a patch baseline:

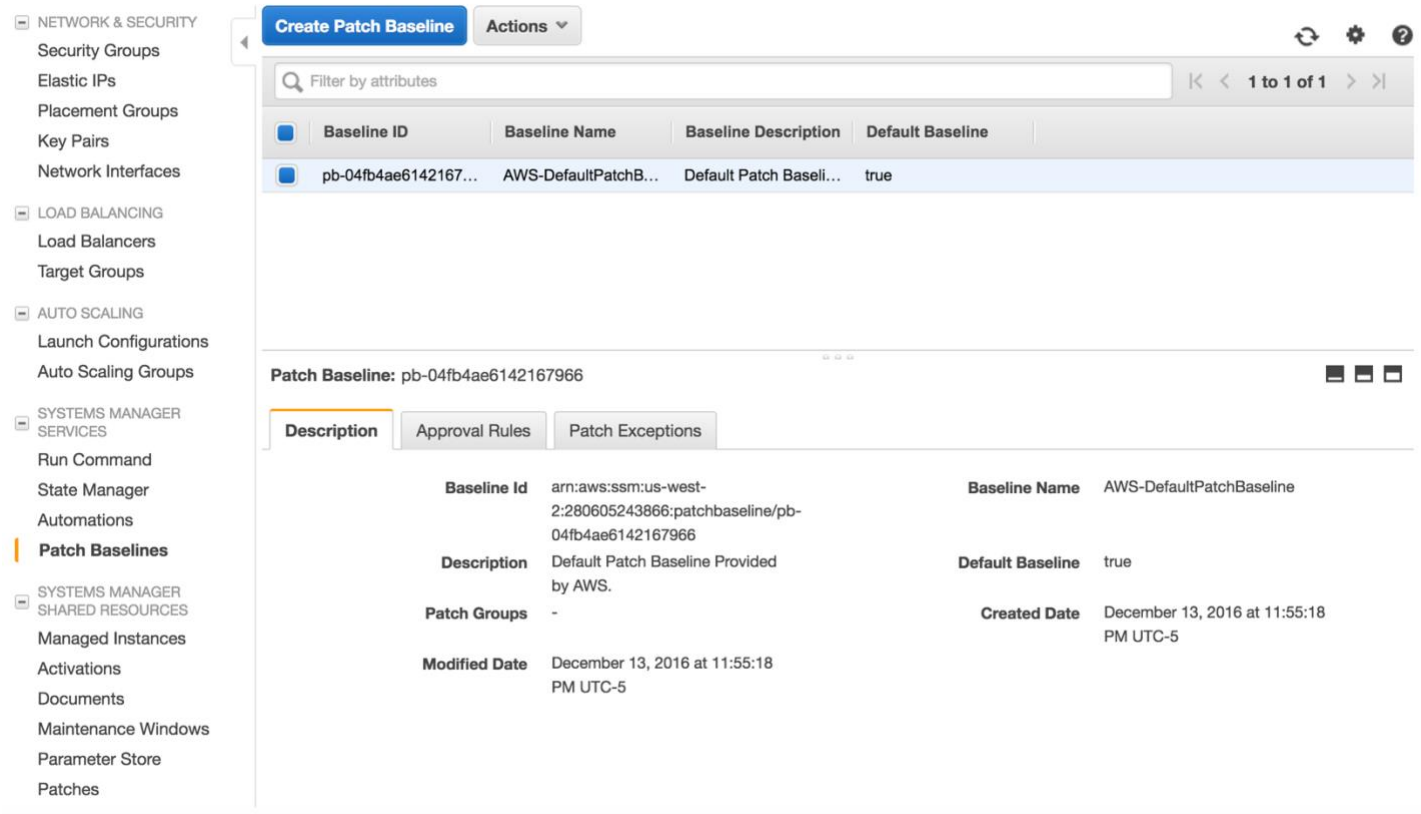
1. On the AWS services page, click **EC2**.



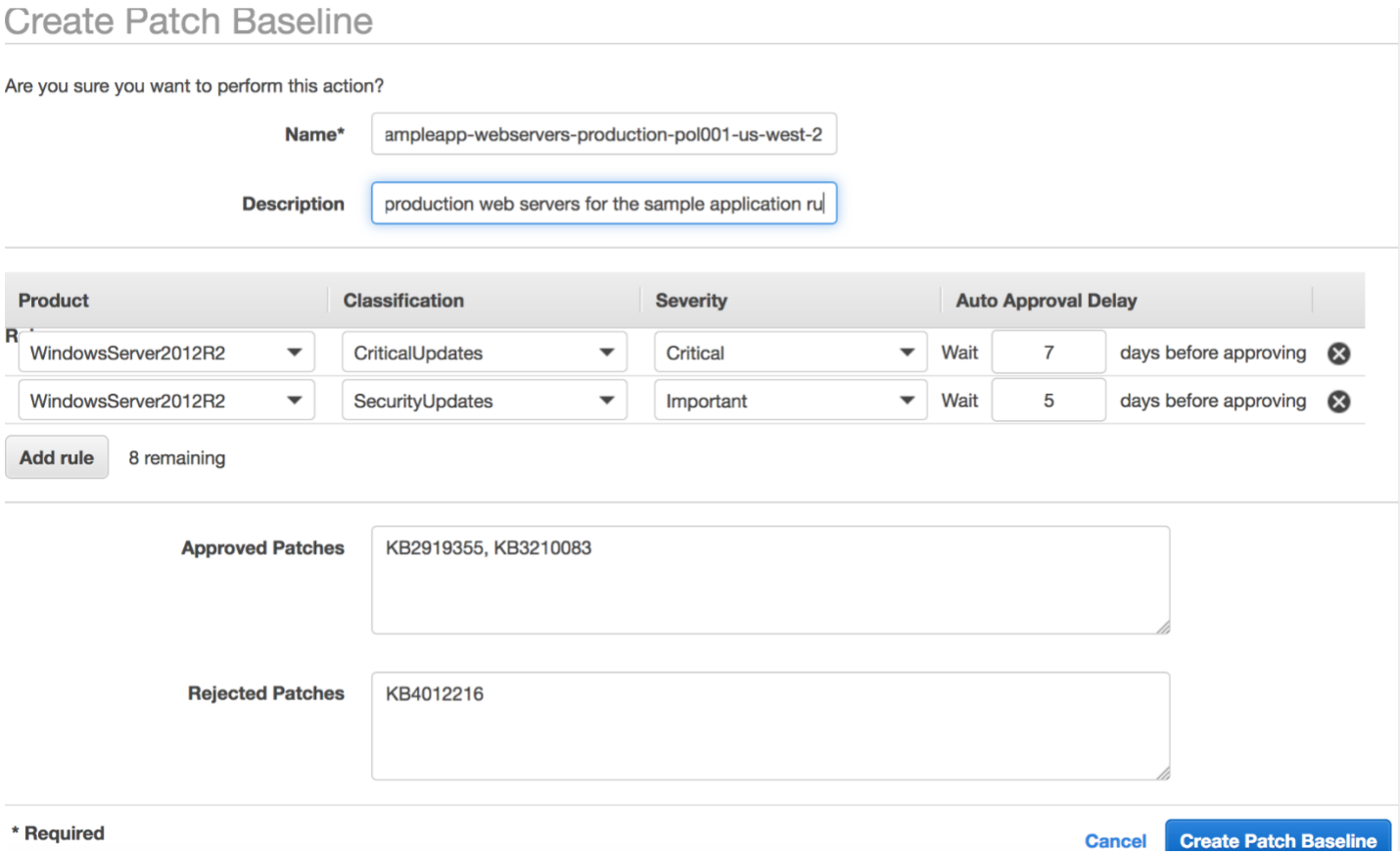
2. Under **SYSTEMS MANAGER SERVICES** on the left, click **Patch Baselines**.



3. Click **Create Patch Baseline**.



4. Fill in the form as follows:



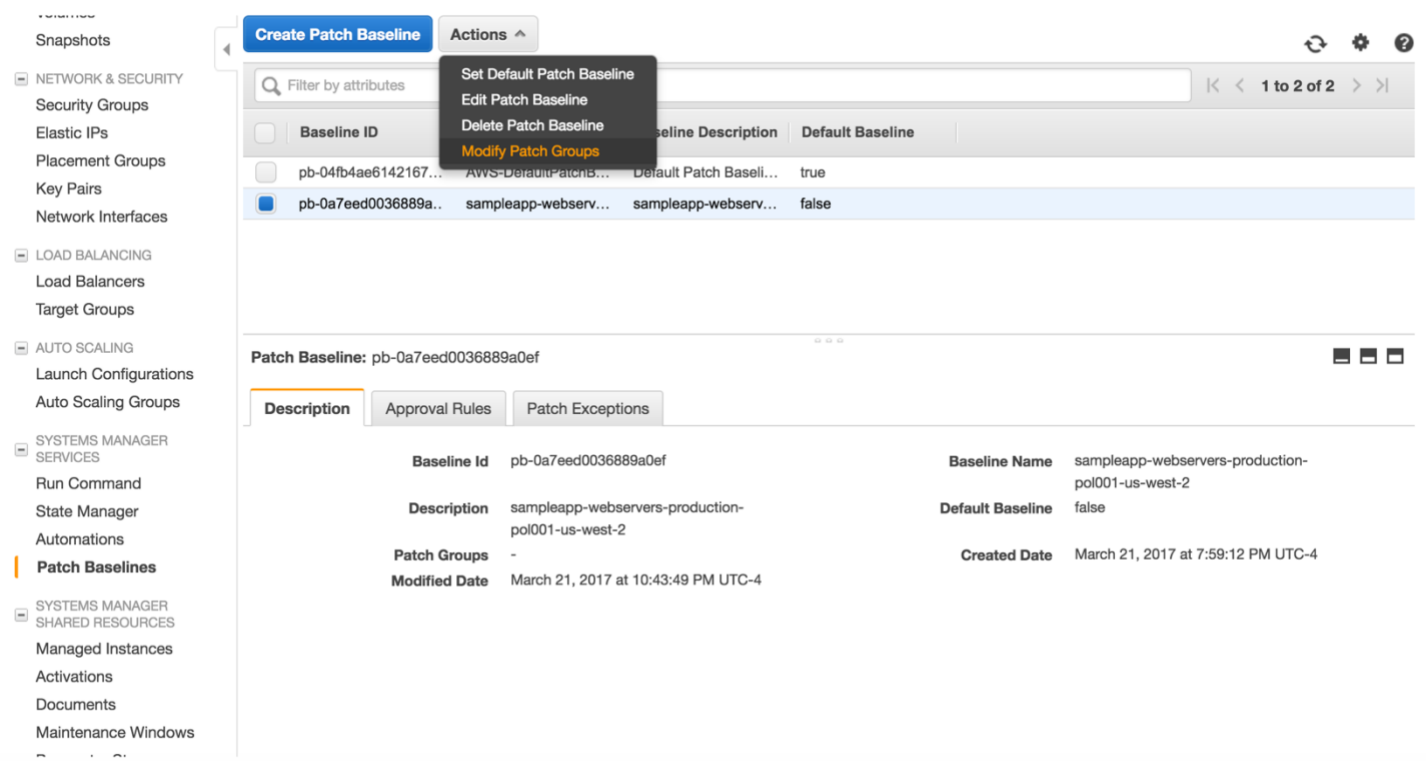
5. Click **Create Patch Baseline**.

Note: See updates at http://www.catalog.update.microsoft.com/Search.aspx.

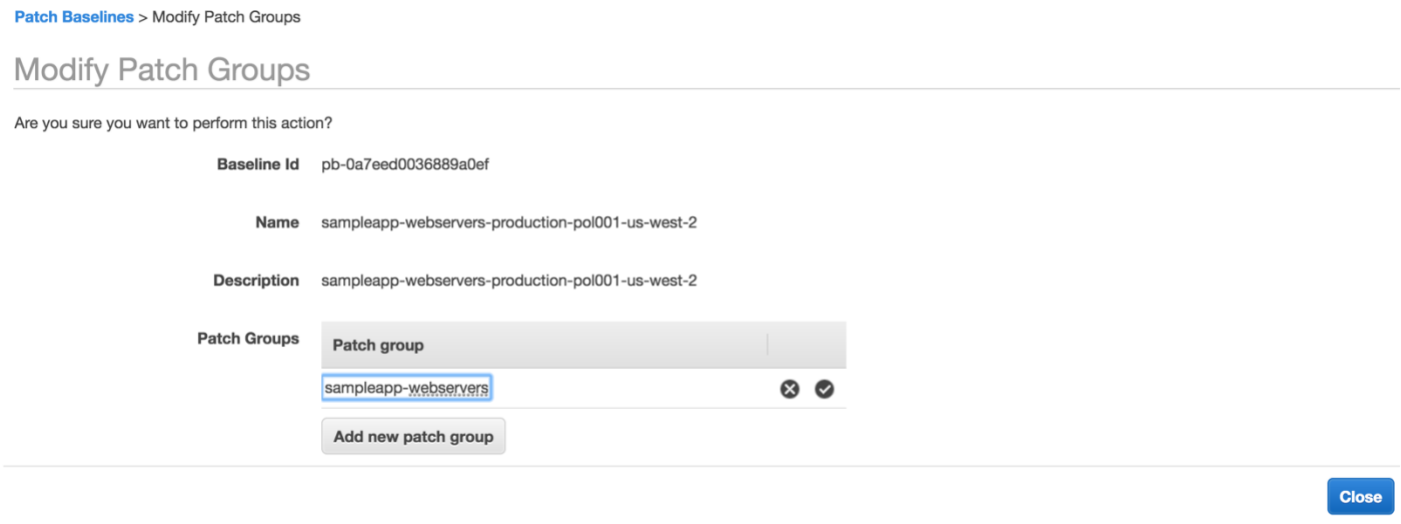
Attaching a Patch Group to the Patch Baseline

After you have the correct patch baseline, you need to associate it with a Patch Group.

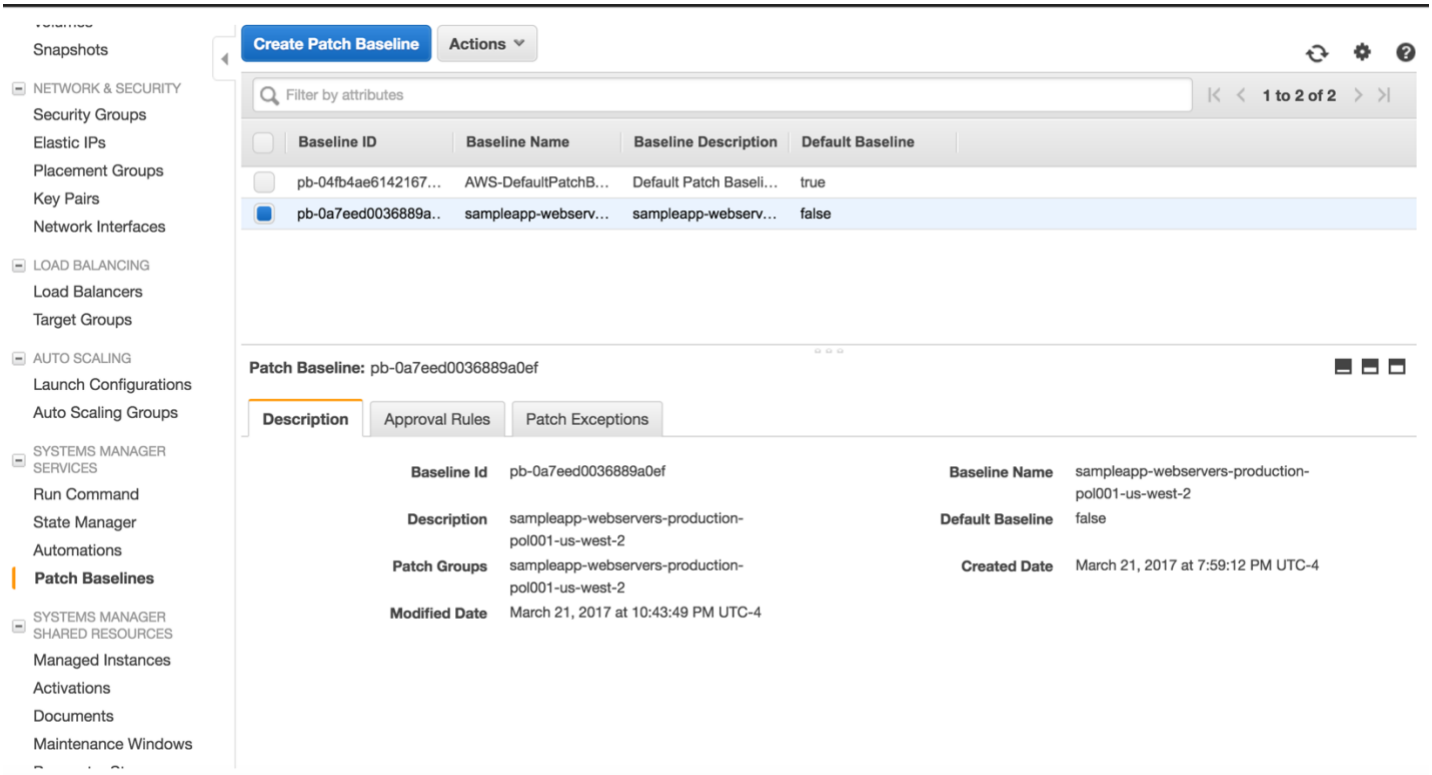
1. On the AWS services page, click **EC2**.
2. Under **SYSTEMS MANAGER SERVICES** on the left, click **Patch Baselines**.
3. Select the patch baseline you created.
4. Click **Actions** and select **Modify Patch Groups**.



5. If you do not see a patch group text box, click **Add new patch group**.
6. In the text box, type **sampleapp-webservers-production-pol001-us-west-2.** Instances that have a tag named Patch Group with this value will have this patch baseline applied to them.
7. Click the checkmark icon to the right, and then click **Close**.



8. On the **EC2** page, click **Patch Baselines**.
9. Click **refresh** (the up/down arrow) and then select your patch policy and ensure that the Patch Groups entry under the Description tag shows your patch group.



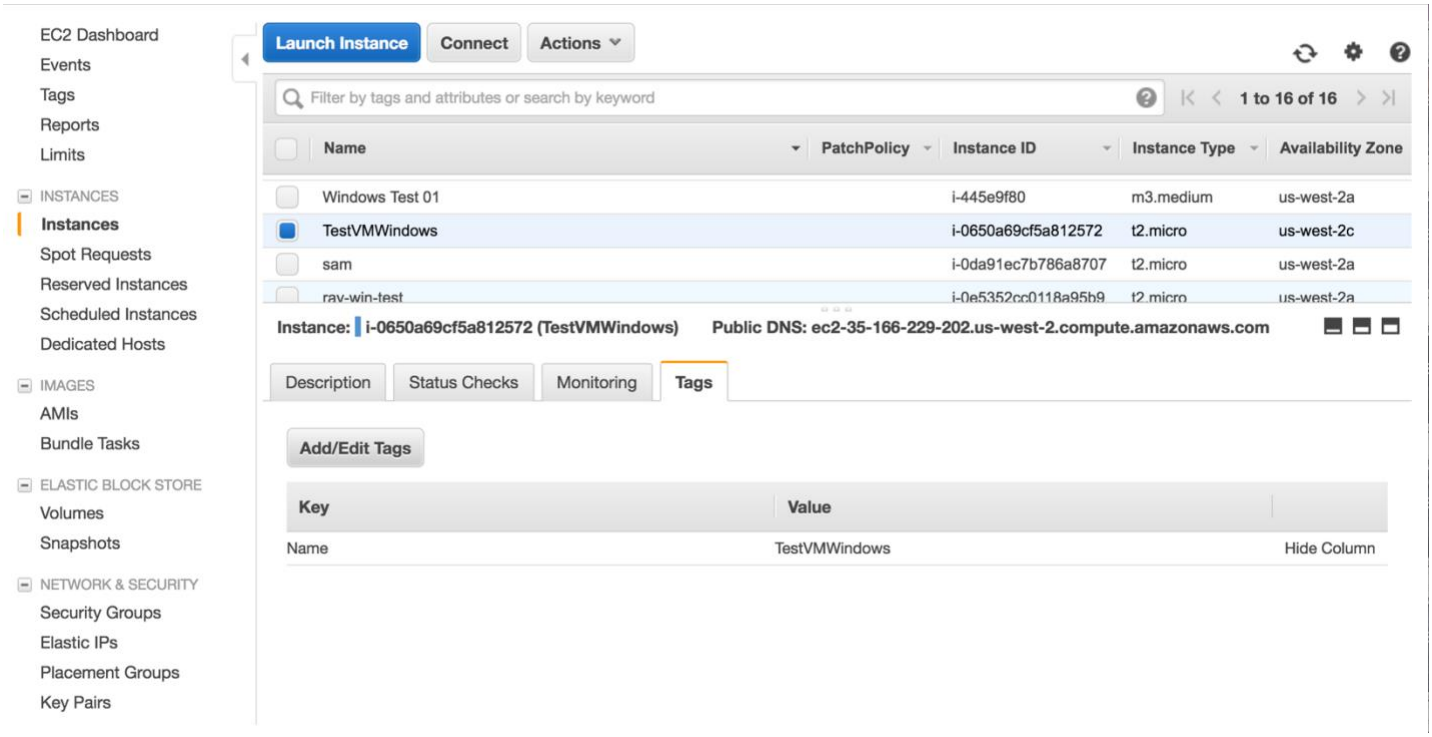
Tagging Windows Instances

Use the Patch Group tag to tag instances. This tag is required for Windows tagging. The tag value is used to pair the instance with the correct patch baseline in a maintenance window. It is assumed that the instances have the ssm agent installed and are active, which is required to run commands on the machine.

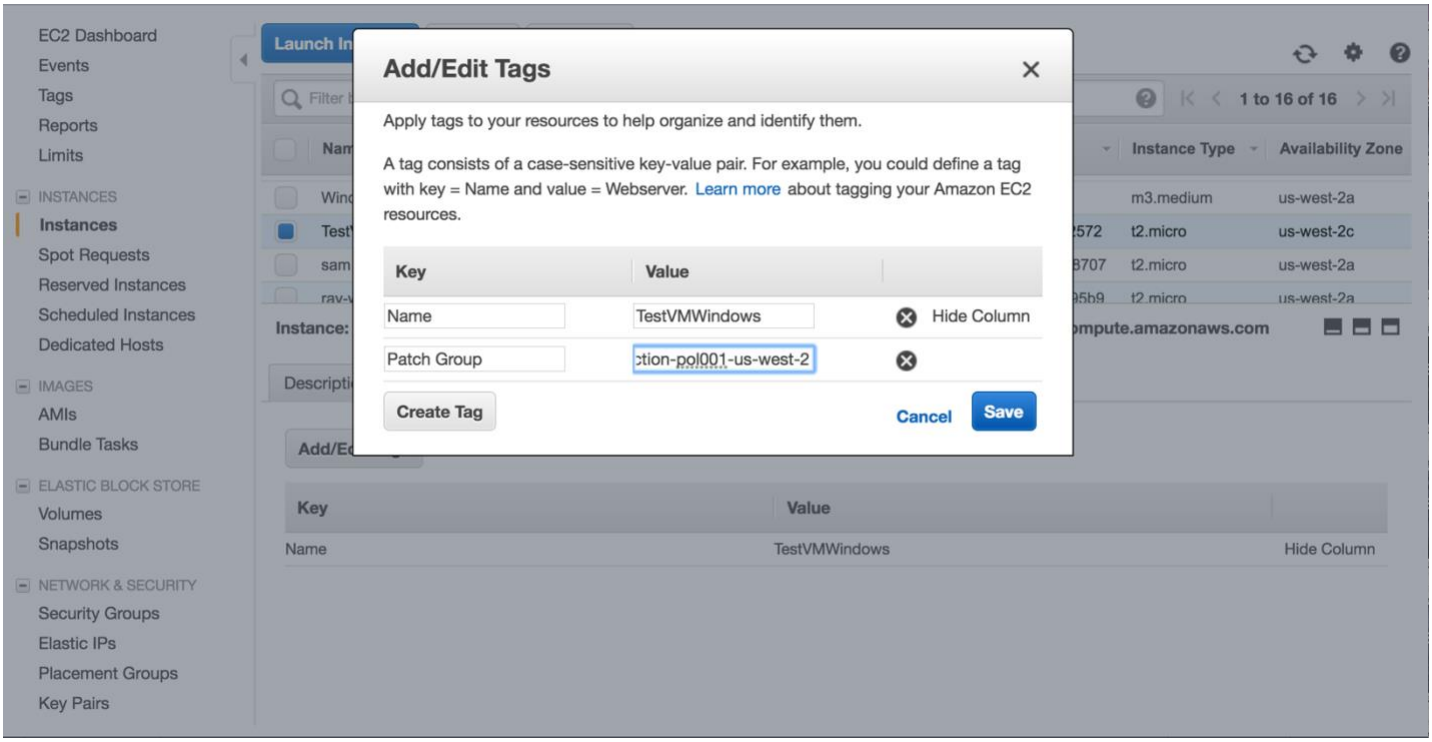
The example shows how to tag an existing instance. You can also set the tag when you create a new instance under a CloudFormation template or start an instance outside of CloudFormation.

Note: You can apply only one patch baseline to a Windows instance.

1. Under **EC2**, find your instance and select it.
2. Click the **Tags** tab below the instance list.

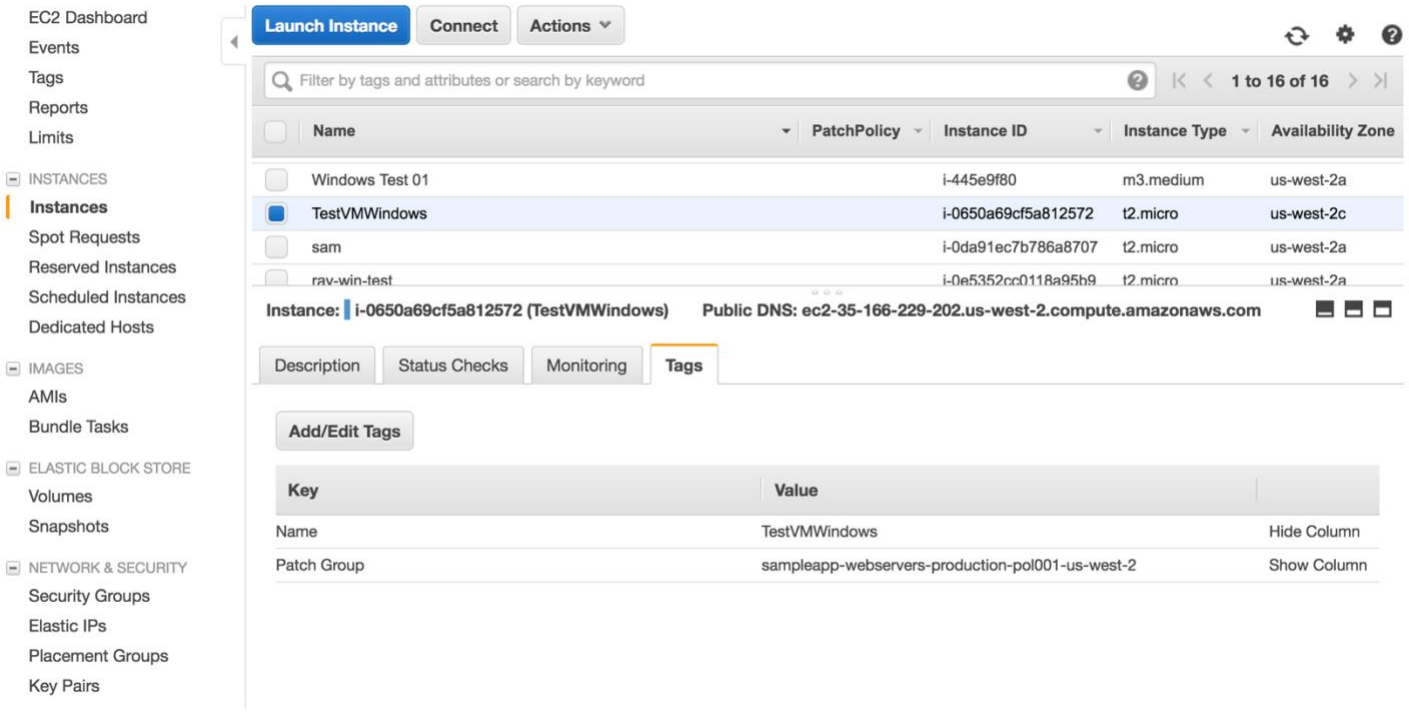


3. Click **Add/Edit Tags**, click **Create Tag** and add a Patch Group tag with a value of **sampleapp-webservers-production-pol001-us-west-2**.



4. Click **Save**.

The instance should look like this:



Creating the Maintenance Window

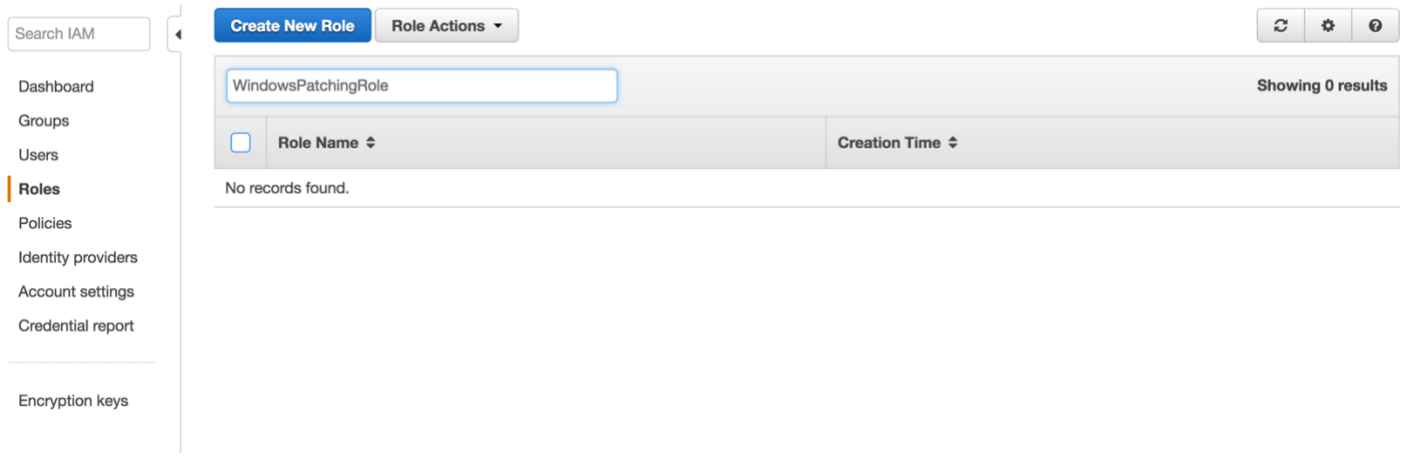
The first step in creating a maintenance window is to create roles to support the maintenance window. The account used to run this maintenance window (your user account) must have the *AmazonSSMFullAccess* policy and the *iam:PassRole* policy.

Note: You can create a separate account for these policies and use that for your maintenance window.

When you create policies and roles, the first step is to create the IAM role that the maintenance window will run under. This role needs the *AmazonSSMMaintenanceWindowRole* policy. By convention, this role is named *WindowsPatchingRole*. The first step is to see if this role exists.

To search for the WindowsPatchingRole:

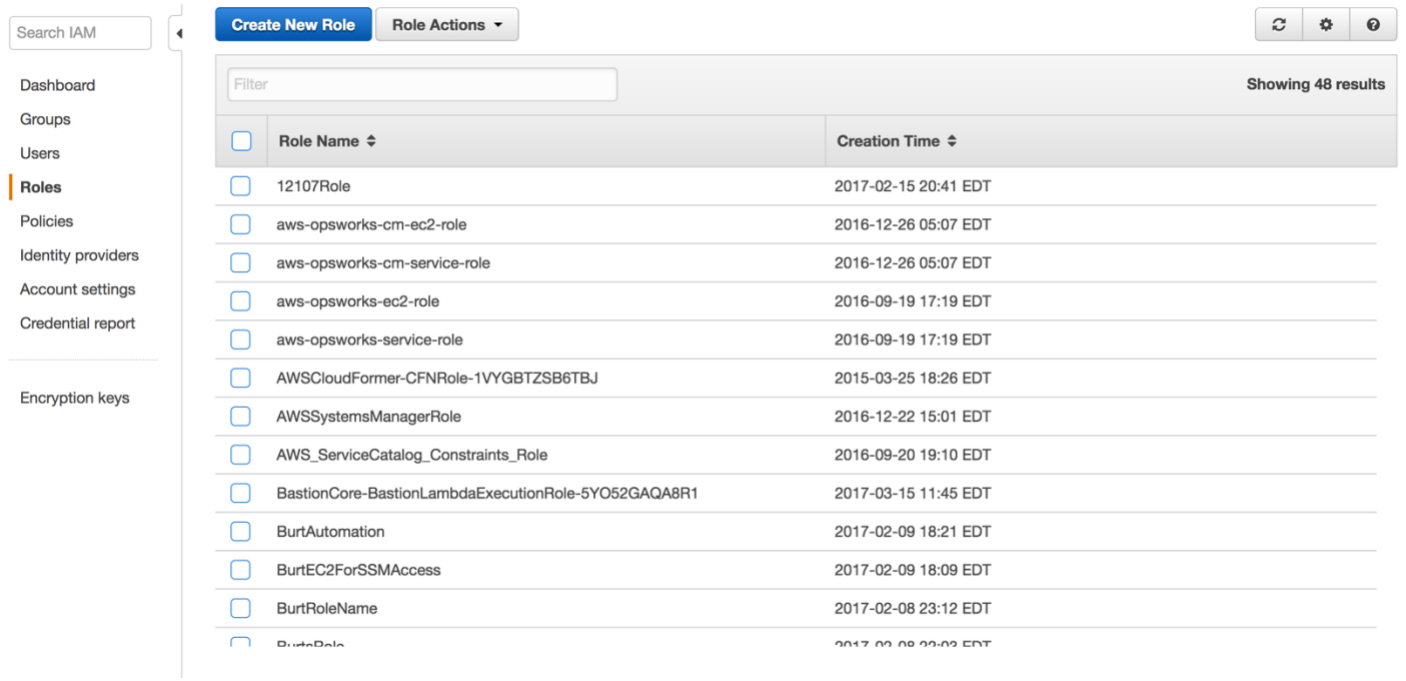
1. Open your AWS console and launch IAM.
2. On the left column, click **Roles**.
3. Type **WindowsPatchingRole** in the search box and press Enter.



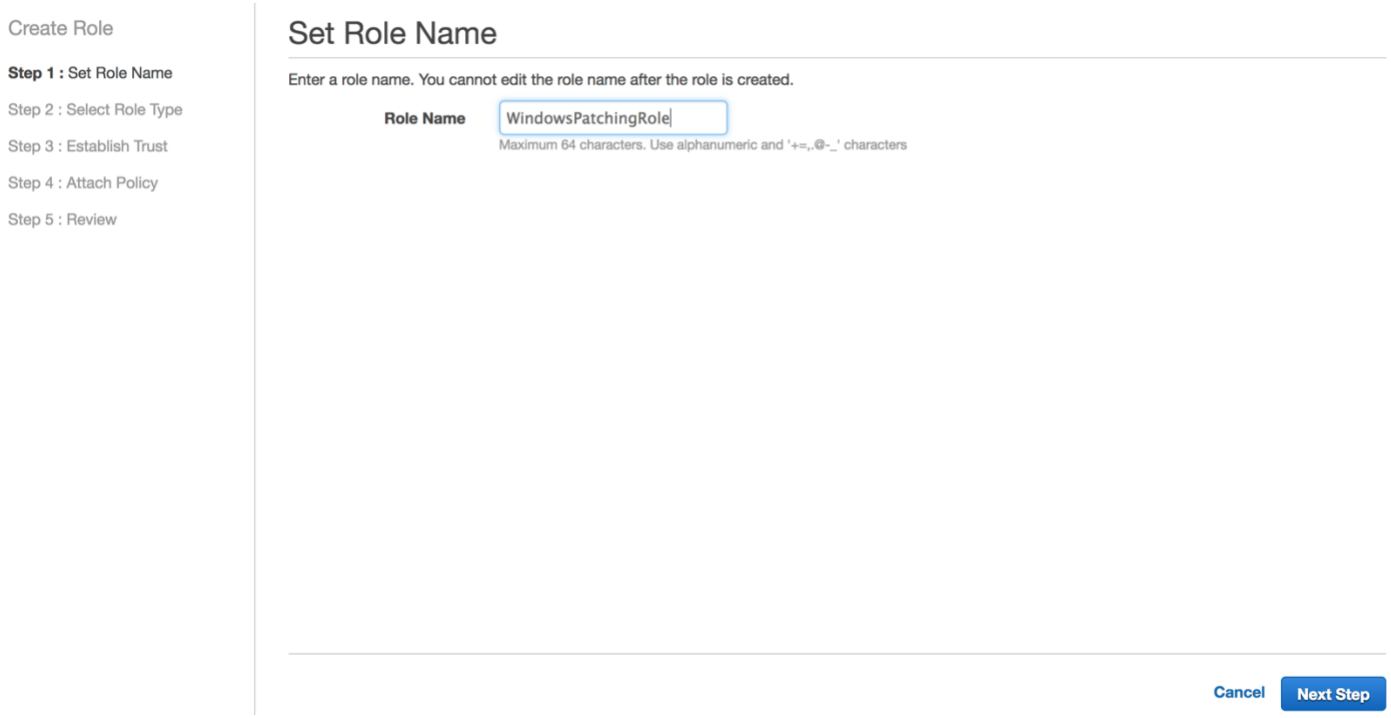
If the role exists, you can skip to the *Configuring Local User Account Permissions* section. If it does not exist, the following steps show how to create it.

To create the WindowsPatchingRole:

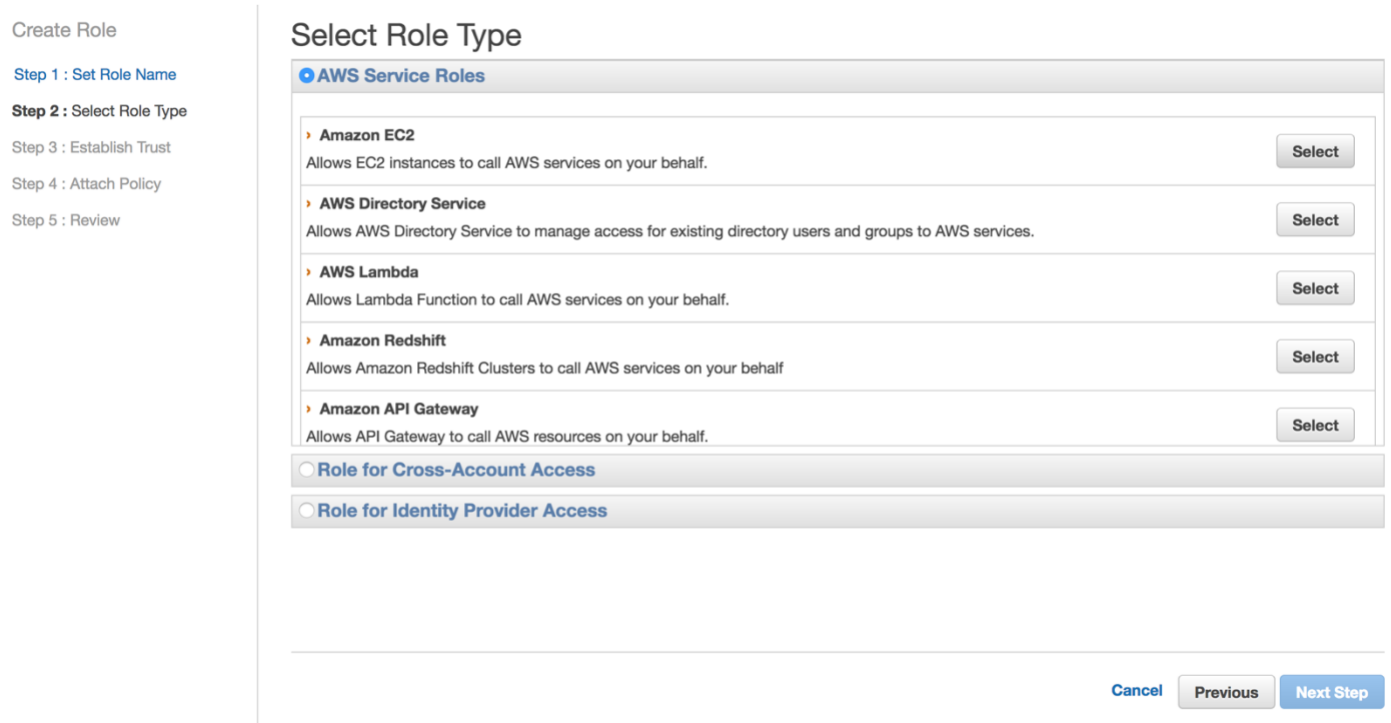
1. In the **IAM** section, click **Roles** on the left.



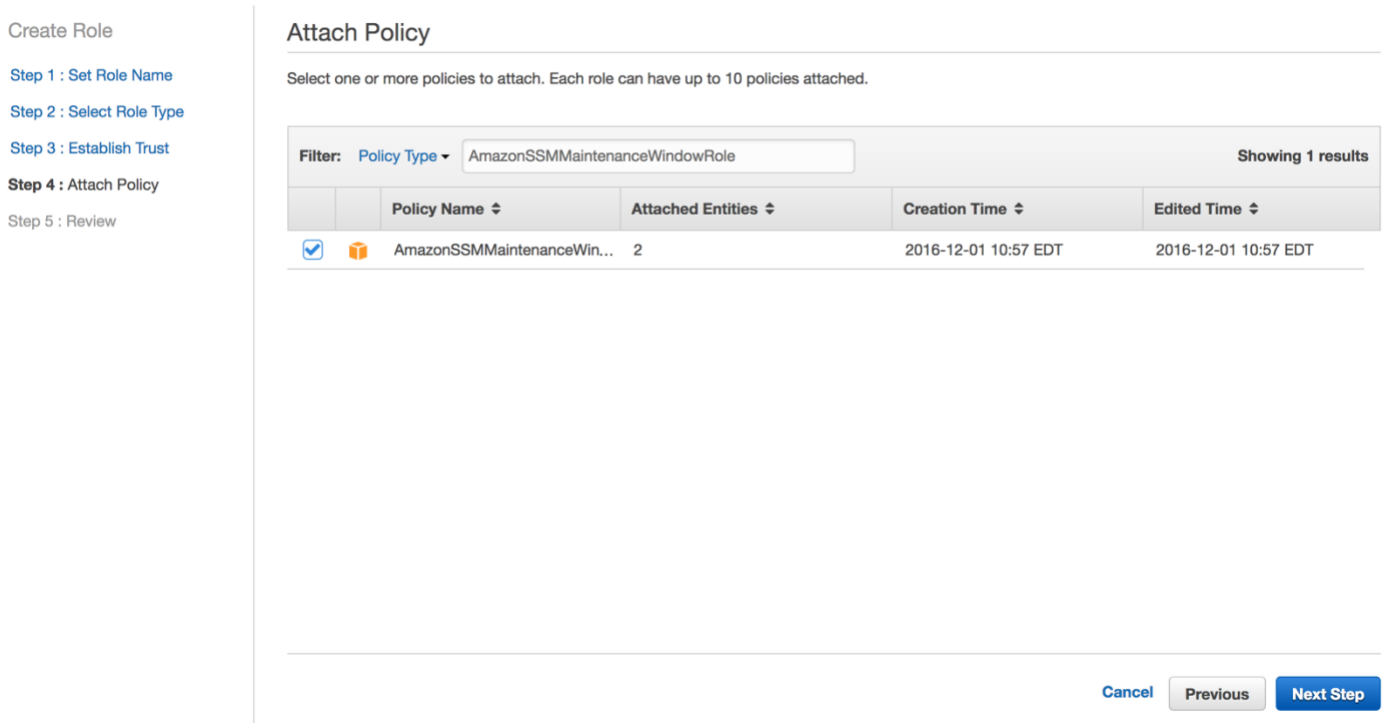
2. Click **Create New Role**.
3. For **Role Name**, type **WindowsPatchingRole**.



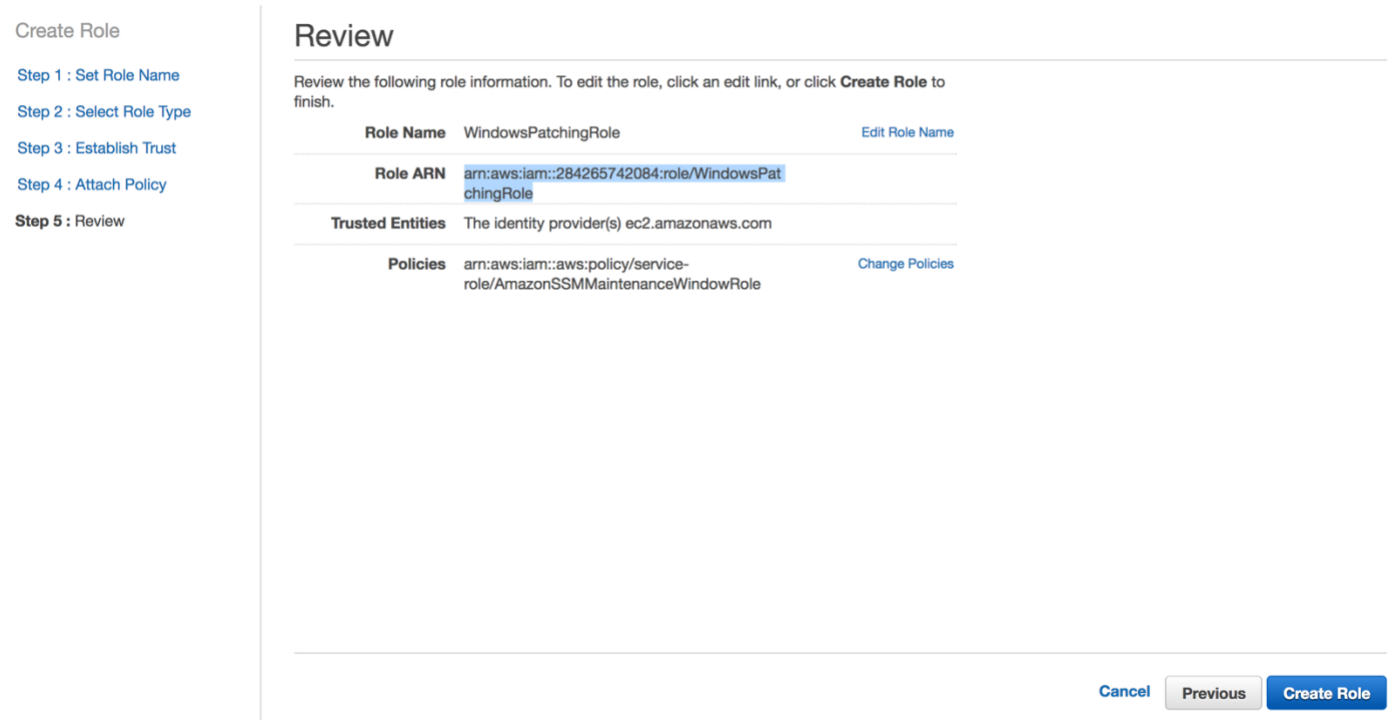
4. Click **Next Step**.
5. Select the **Amazon EC2** role under **AWS Service Role** and click **Next Step**.



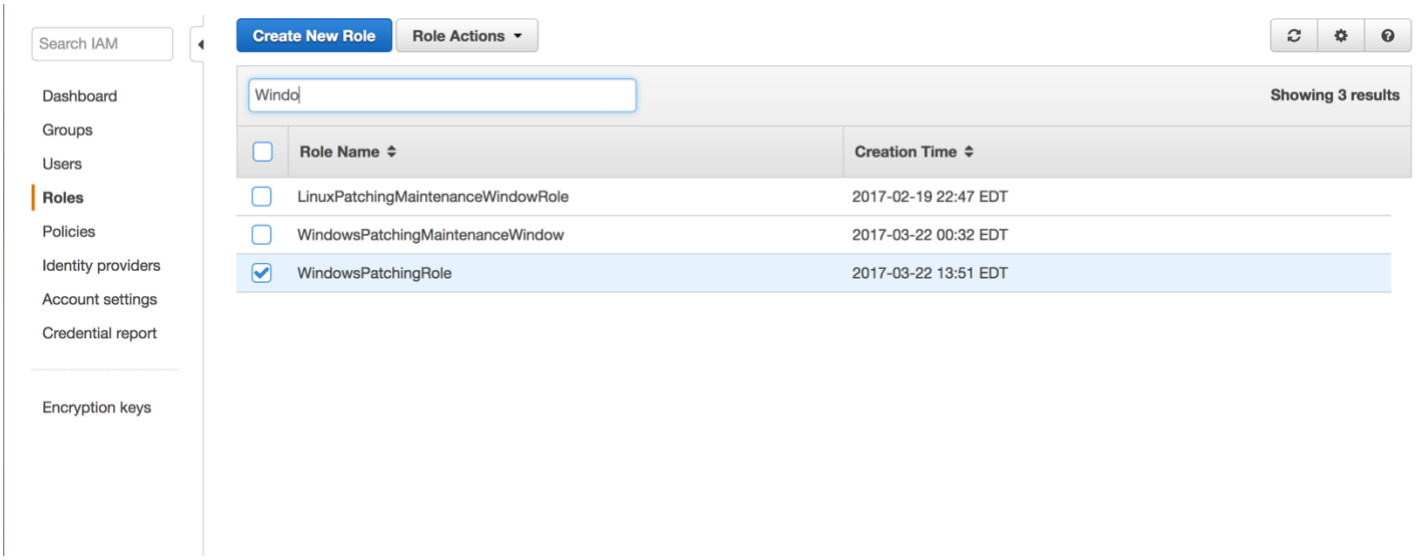
6. Type **AmazonSSMMaintenanceWindowRole** in the search window and select this policy.



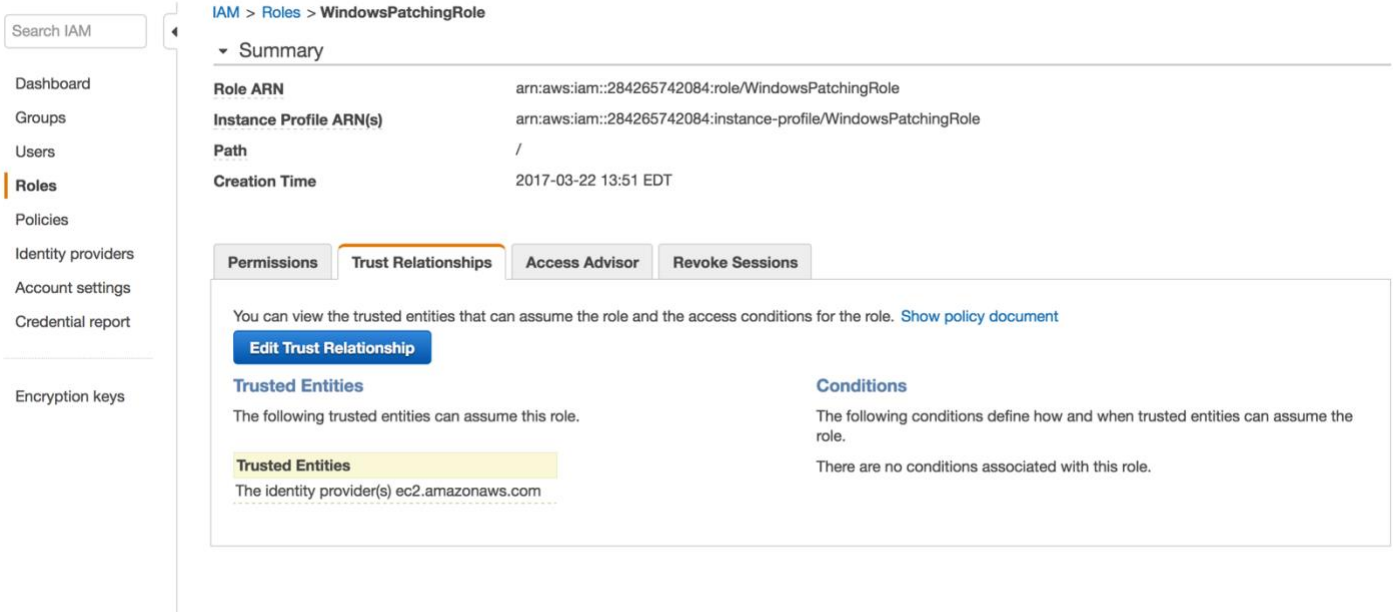
7. Click **Next Step**.
8. Write down the name of the role and click **Create Role**.



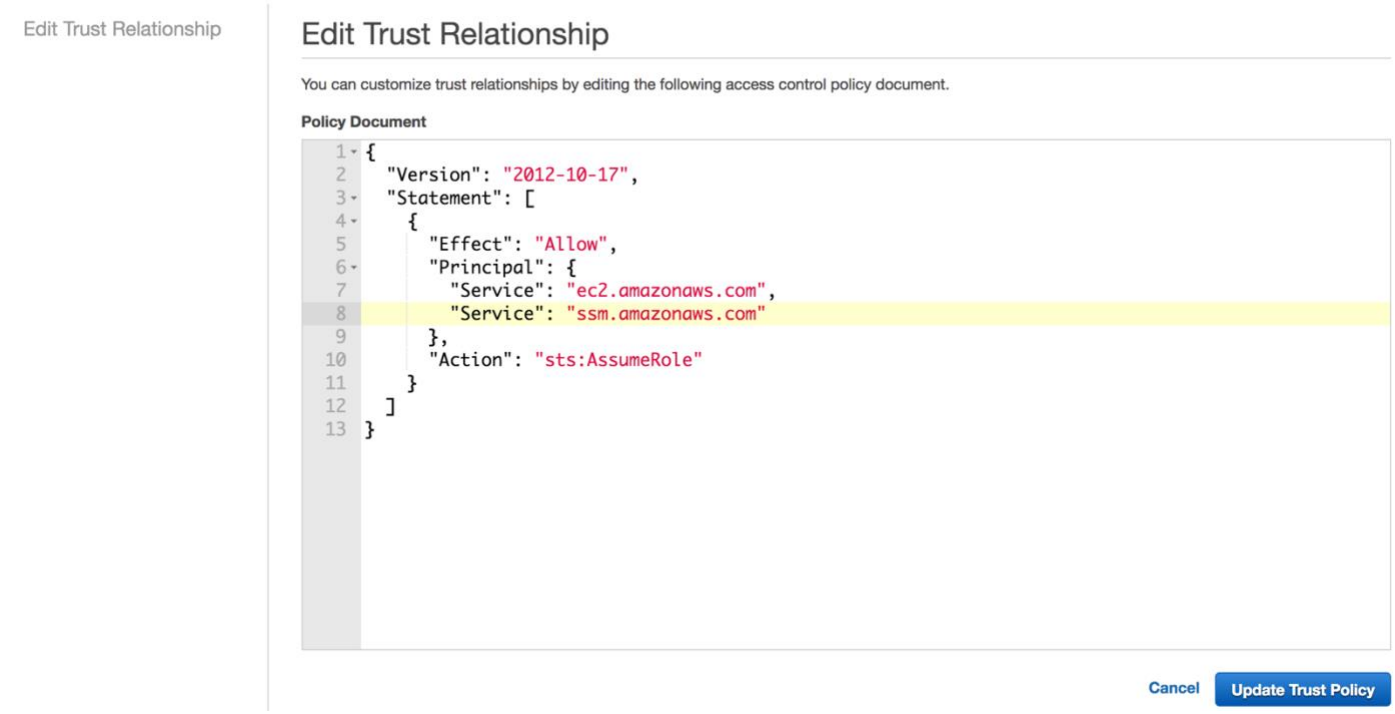
9. Select your newly created role by typing **Windows** in the search box and then selecting your role.



10. On the **Trust Relationships** tab, click **Edit Trust Relationships**.



11. Ensure that **"Service" : "ssm.amazonaws.com"** appears as a Principal attribute. If it does not exist, add it as an attribute.

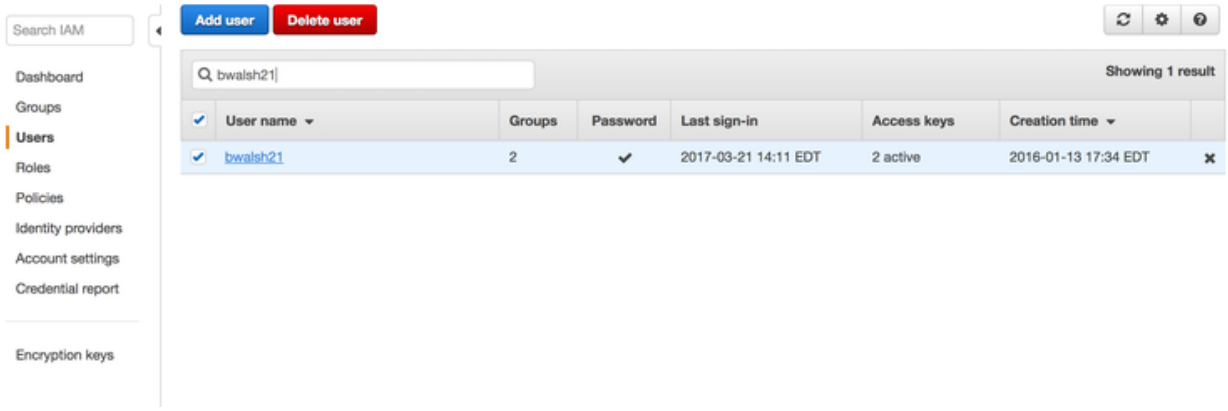


12. Click **Update Trust Policy**.

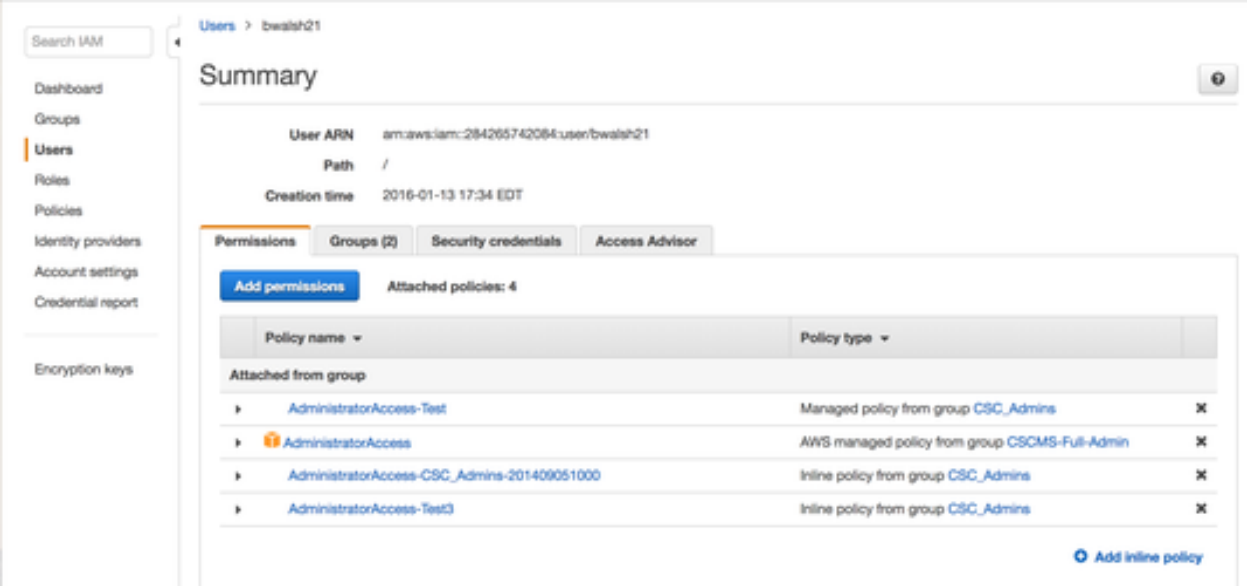
Configuring Local User Account Permissions

To configure your local user account permissions:

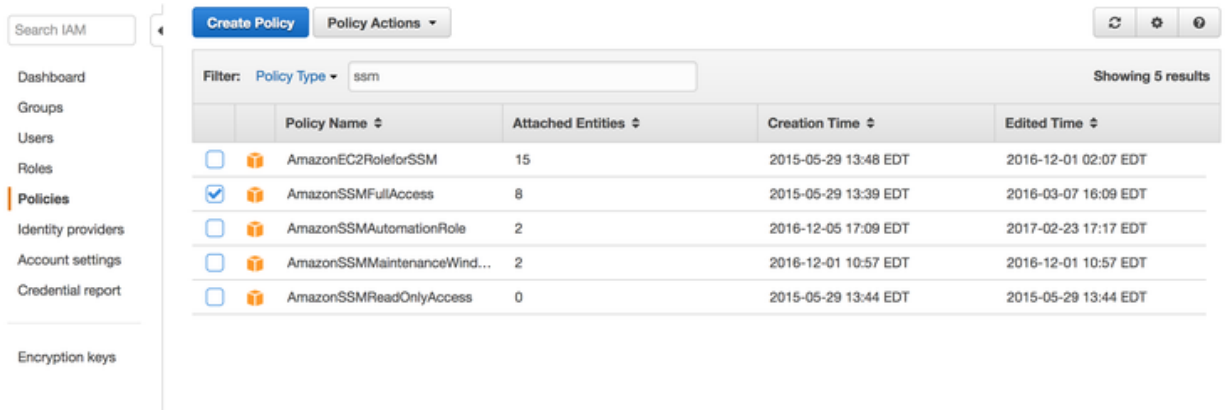
1. Under **IAM**, click **Users** on the left.



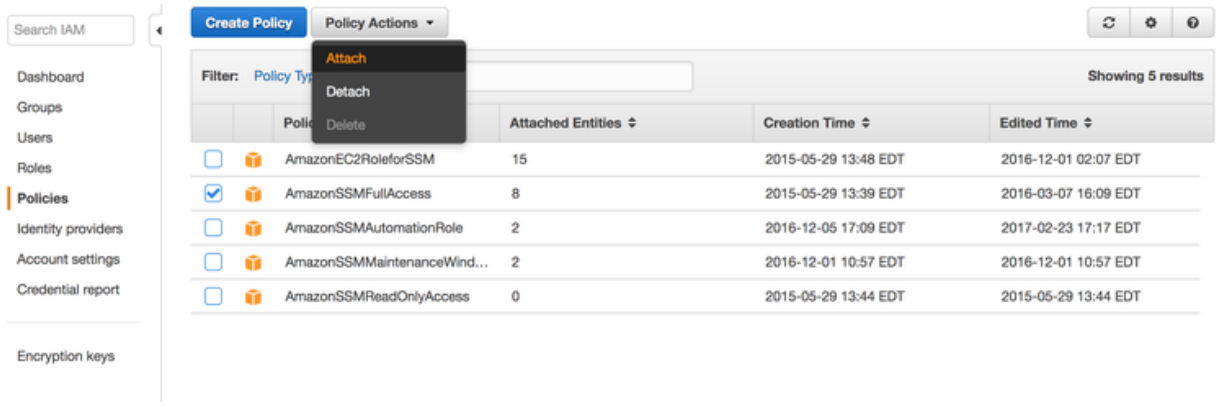
2. In the search box, type your username.
3. Under your user account, click the **Permissions** tab on the right and see if the **AmazonSSMFullAccess** policy is attached to your account.



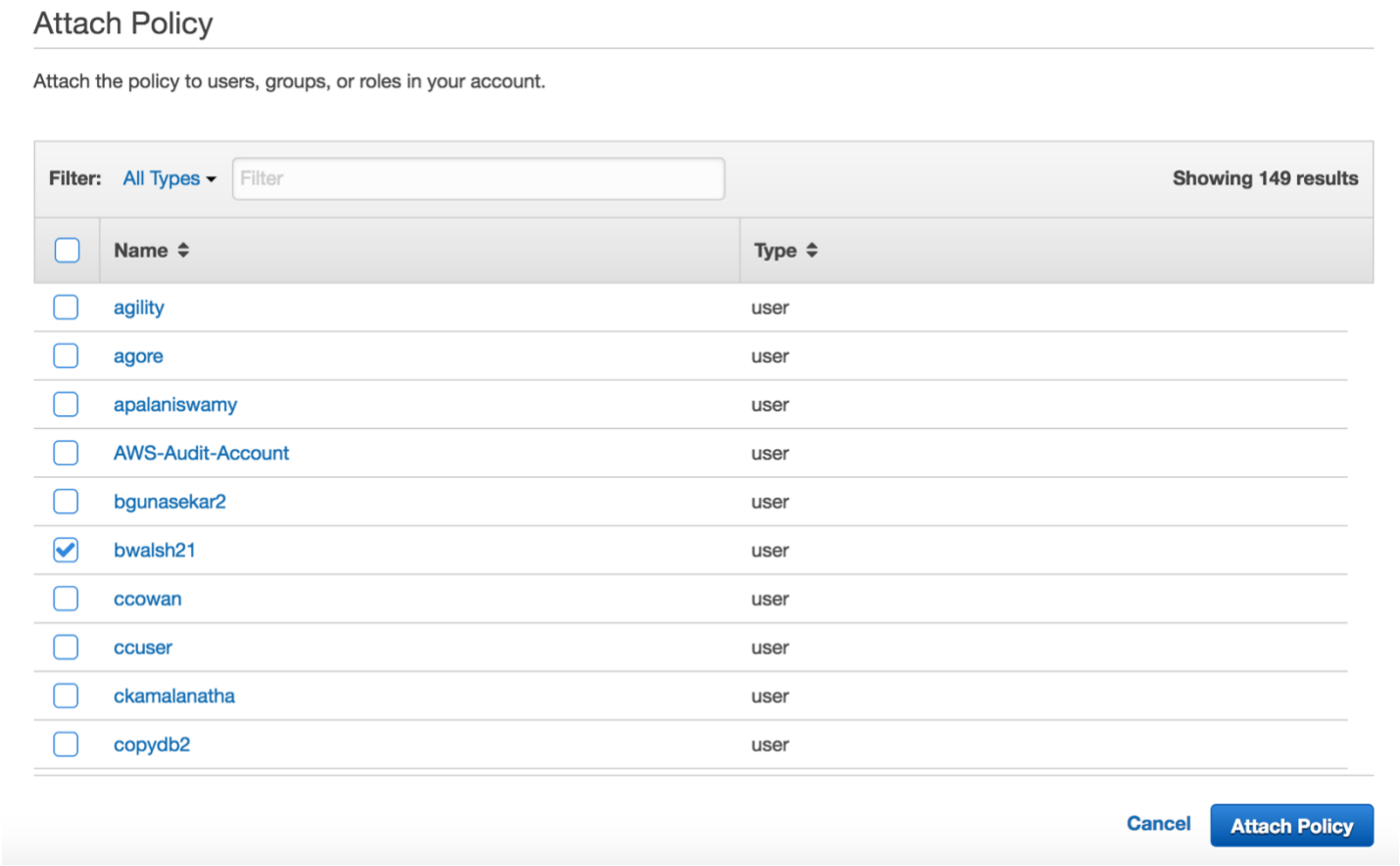
4. If the policy is attached, skip to the step that starts "Go back to IAM" and find and select the **AmazonSSMFullAccess** policy.
5. If the policy is not attached, click **Policies** on the left.



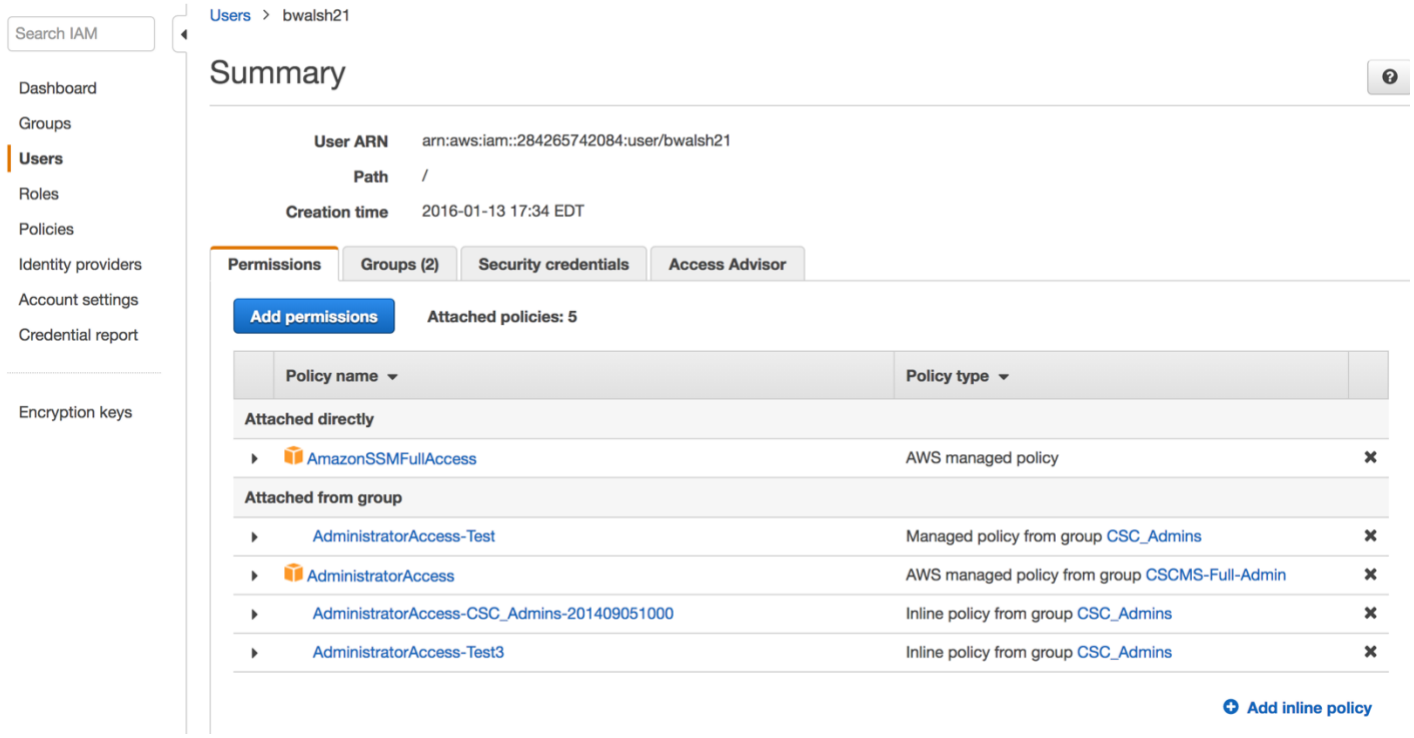
6. Select the **AmazonSSMFullAccess** policy
7. Click **Policy Action** and select **Attach**.



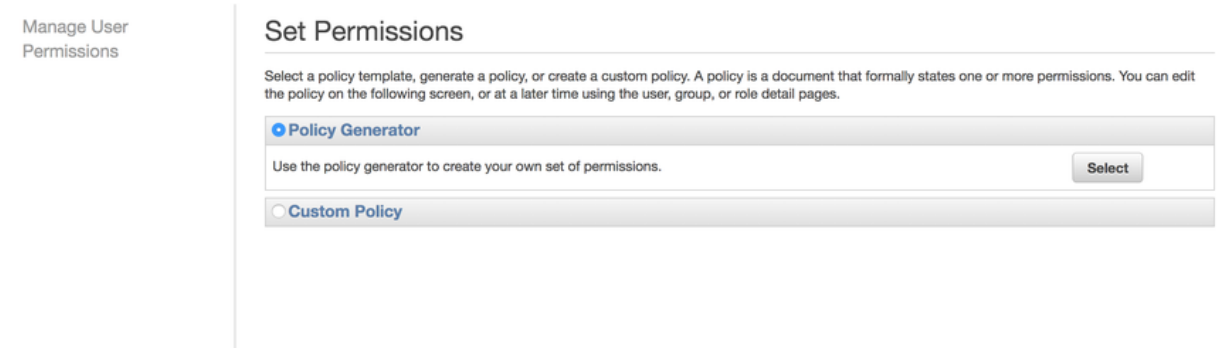
8. Under **Attach Policy**, select your user account and click **Attach Policy**.



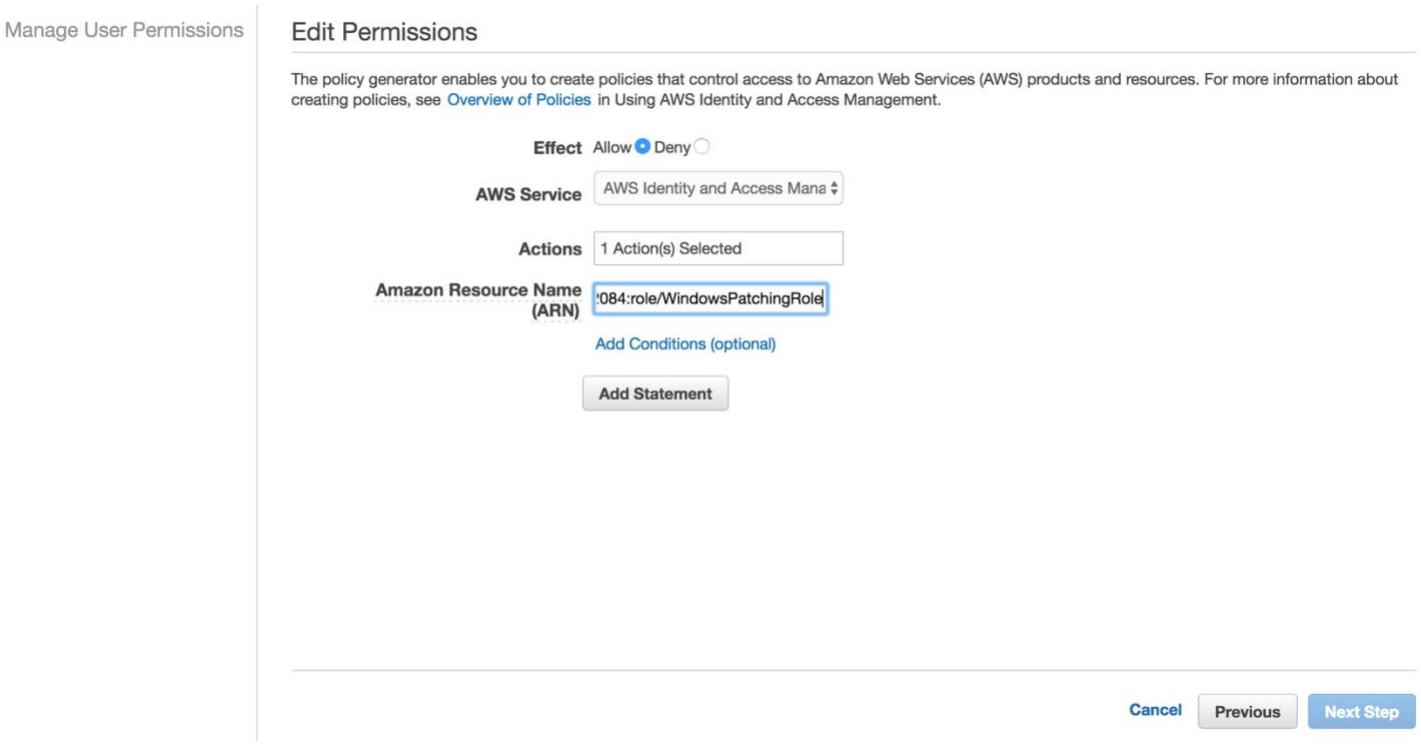
9. Go back to **IAM** > **Users**, find your user ID, and click **Add inline policy** on the bottom right.



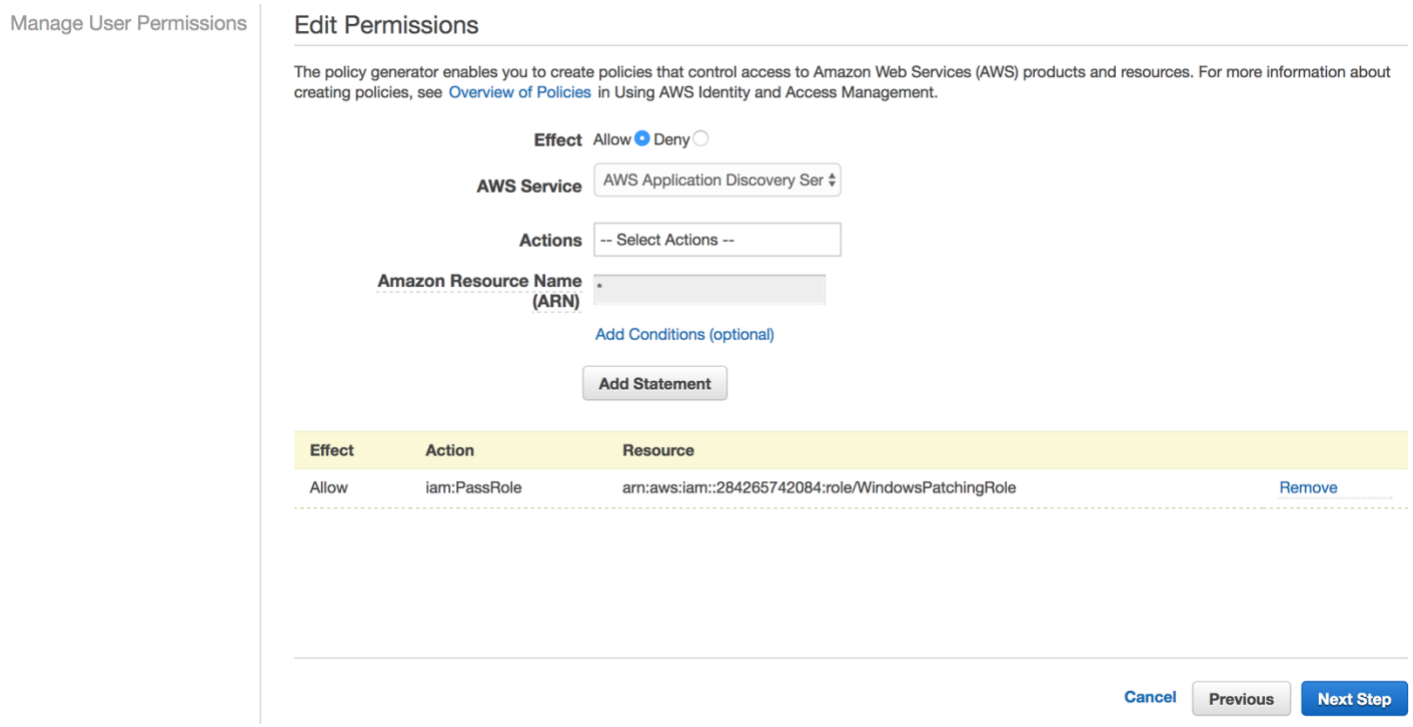
10. Under **Set Permissions**, select **Policy Generator** and click **Select**.



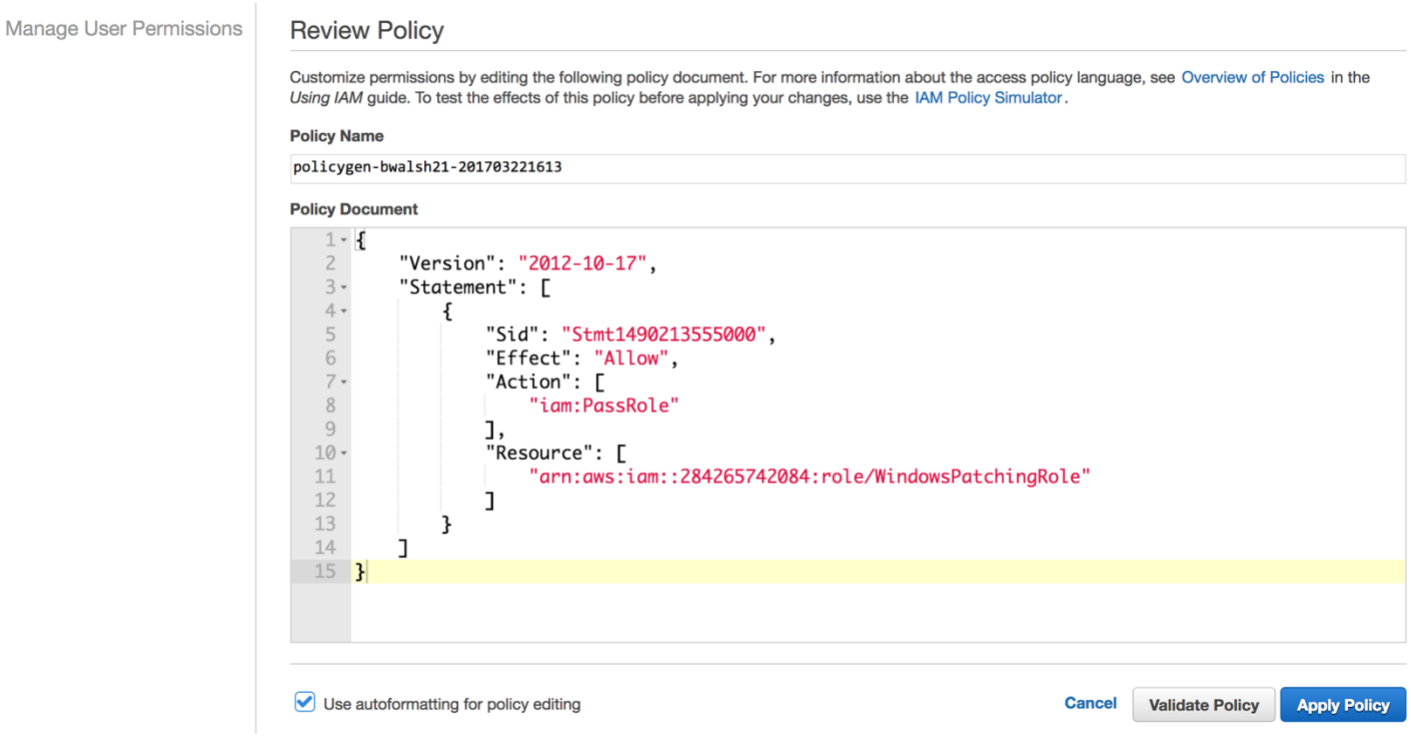
11. On the **Edit Permissions** page, select the following:
    - For **Effect**, select **Allow**
    - For **AWS Service**, select **AWS Identity and Access Management**
    - For **Actions**, select **Pass Role**
    - For **Amazon Resource Name (ARN)**, type arn:aws:iam::284265742084:role/WindowsPatchingRole



12. Click **Add Statement**.
13. Click **Next Step**.



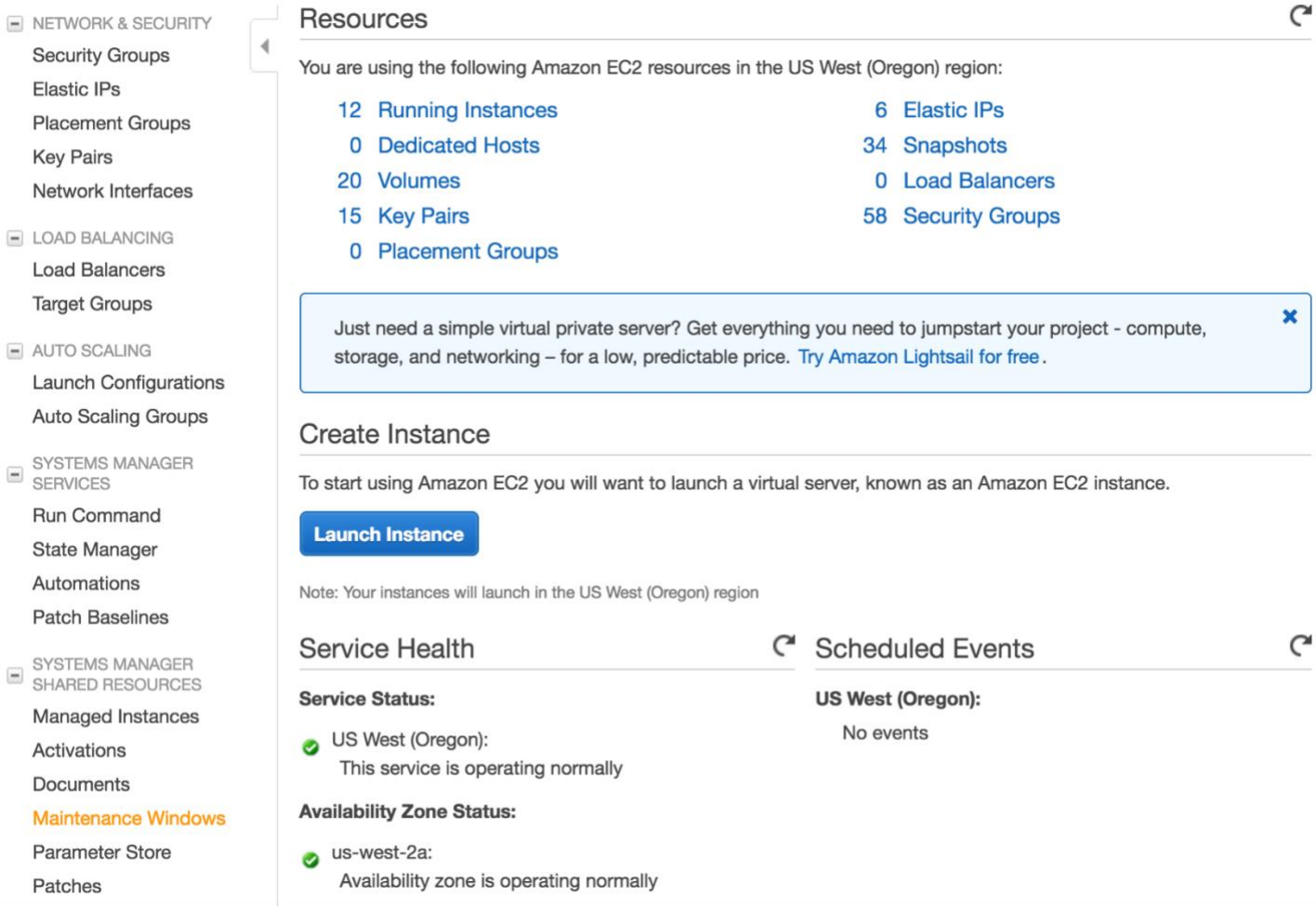
14. Under **Review Policy**, click **Apply Policy**.



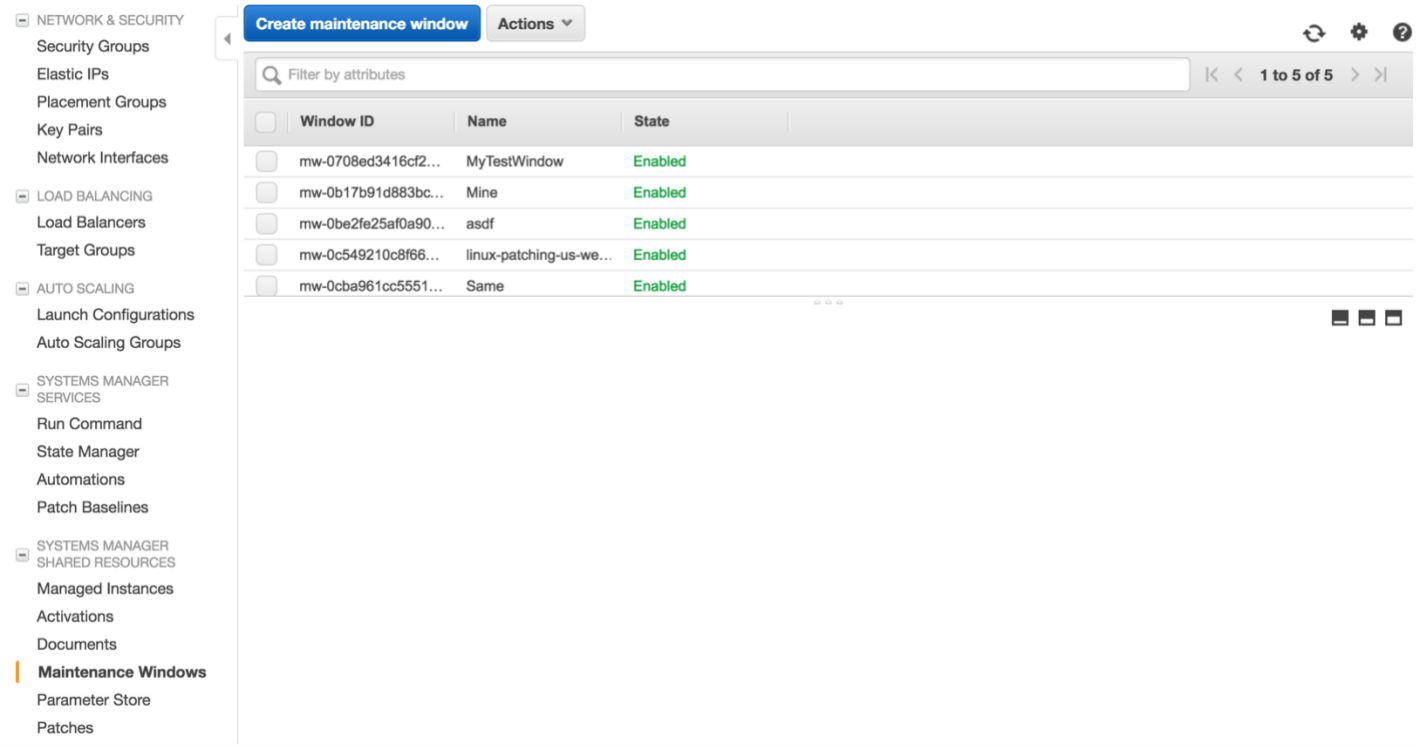
Creating a Maintenance Window

To create a maintenance window

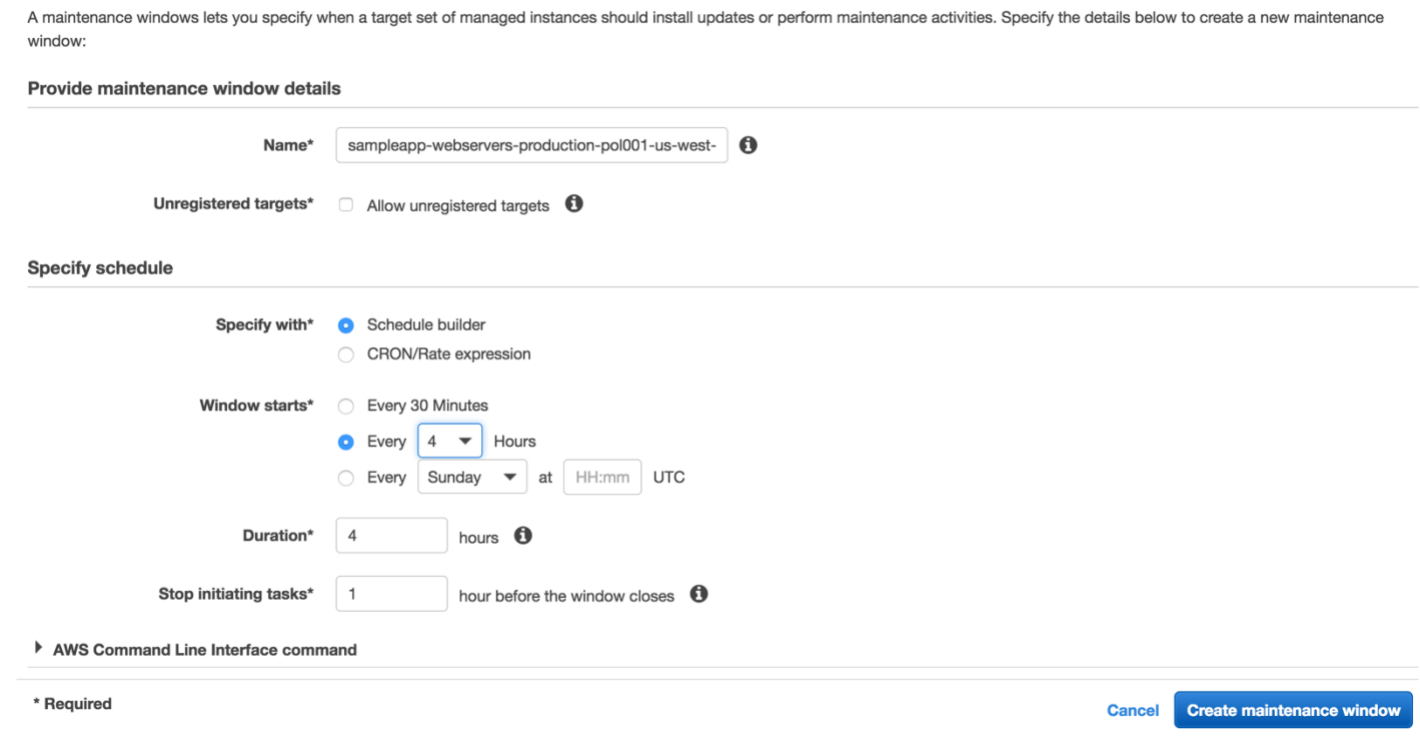
1. Under **EC2**, click **Maintenance Windows** on the left.



2. Click **Create maintenance window**.



3. For **Name**, enter **sampleapp-webservers-production-pol001-us-west-2**.



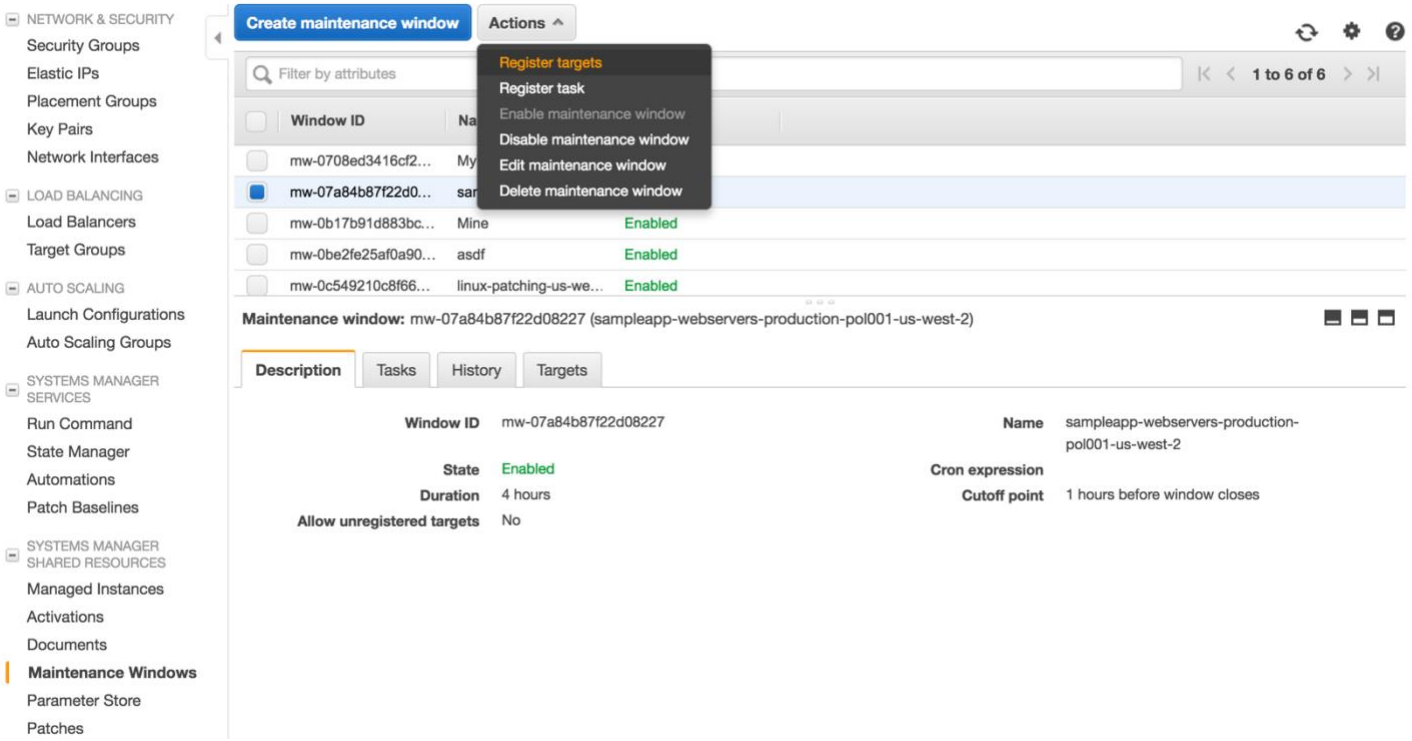
4. For **Window starts**, select how often the maintenance window starts.

5. Select values for **Duration** and **Stop initiating tasks**.

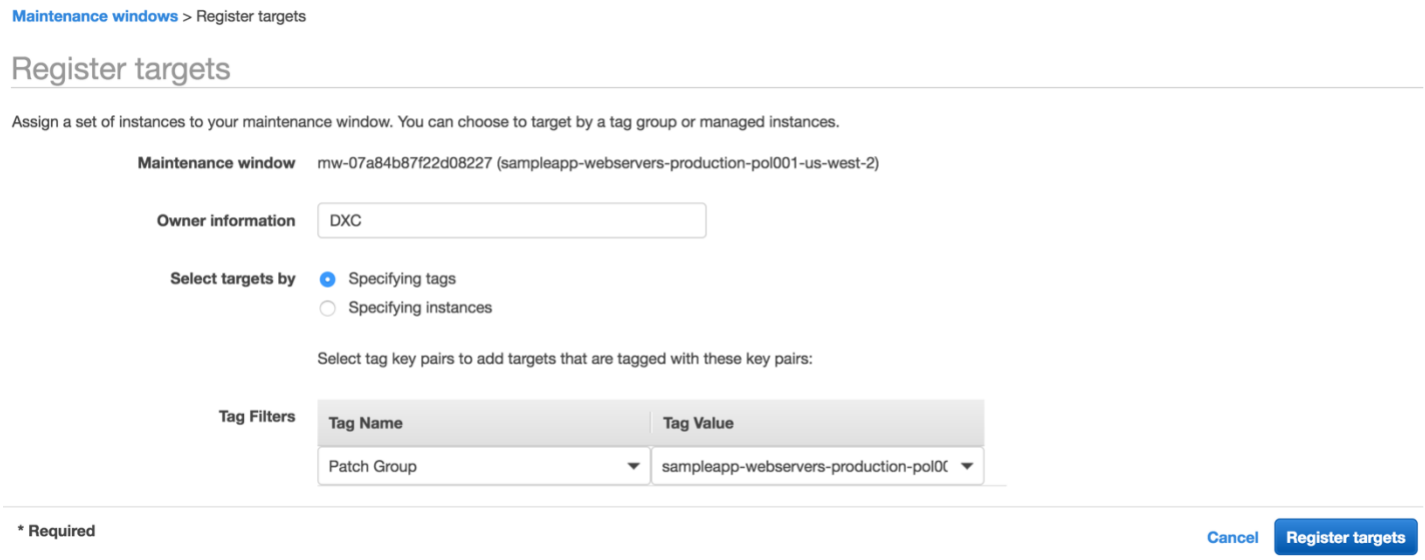
Registering Targets

Register the targets for the maintenance window. The targets are the instances with the Tag Patch Group that have the value **sampleapp-webservers-production-pol001-us-west-2**, which is the value associated with your patch baseline.

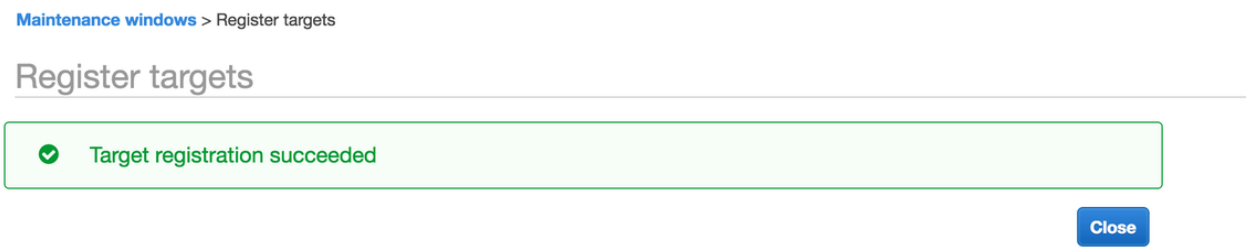
1. Under EC2, click **Maintenance Windows** on the left.
2. On the right, select your maintenance window.
3. Click **Actions** and select **Register targets**.



4. For **Owner information**, enter **DXC**.



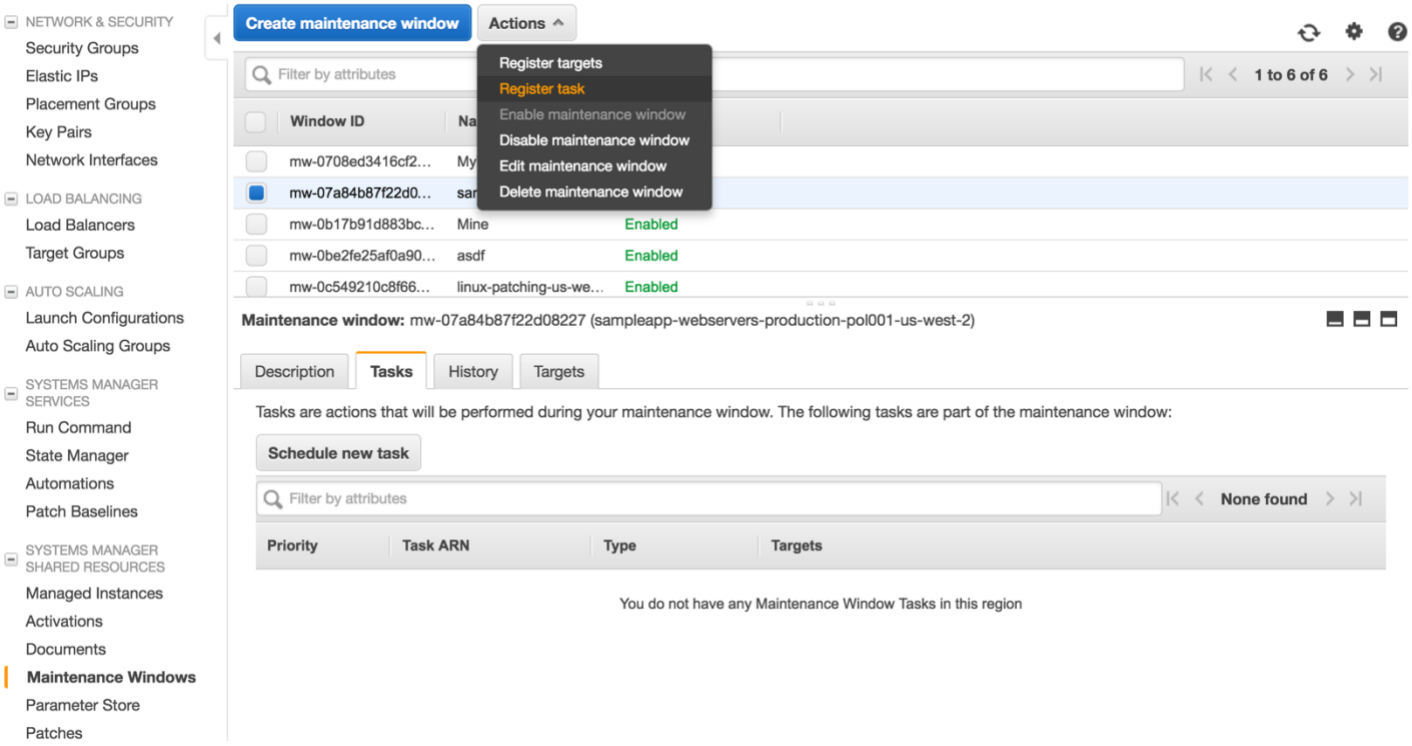
5. For **Select targets by**, click **Specifying tags**.
6. For **Tag Filters**, set the **Tag Name** to Patch Group and the **Tag Value** to **sampleapp-webservers-production-pol001-us-west-2.**
7. Click **Close**.



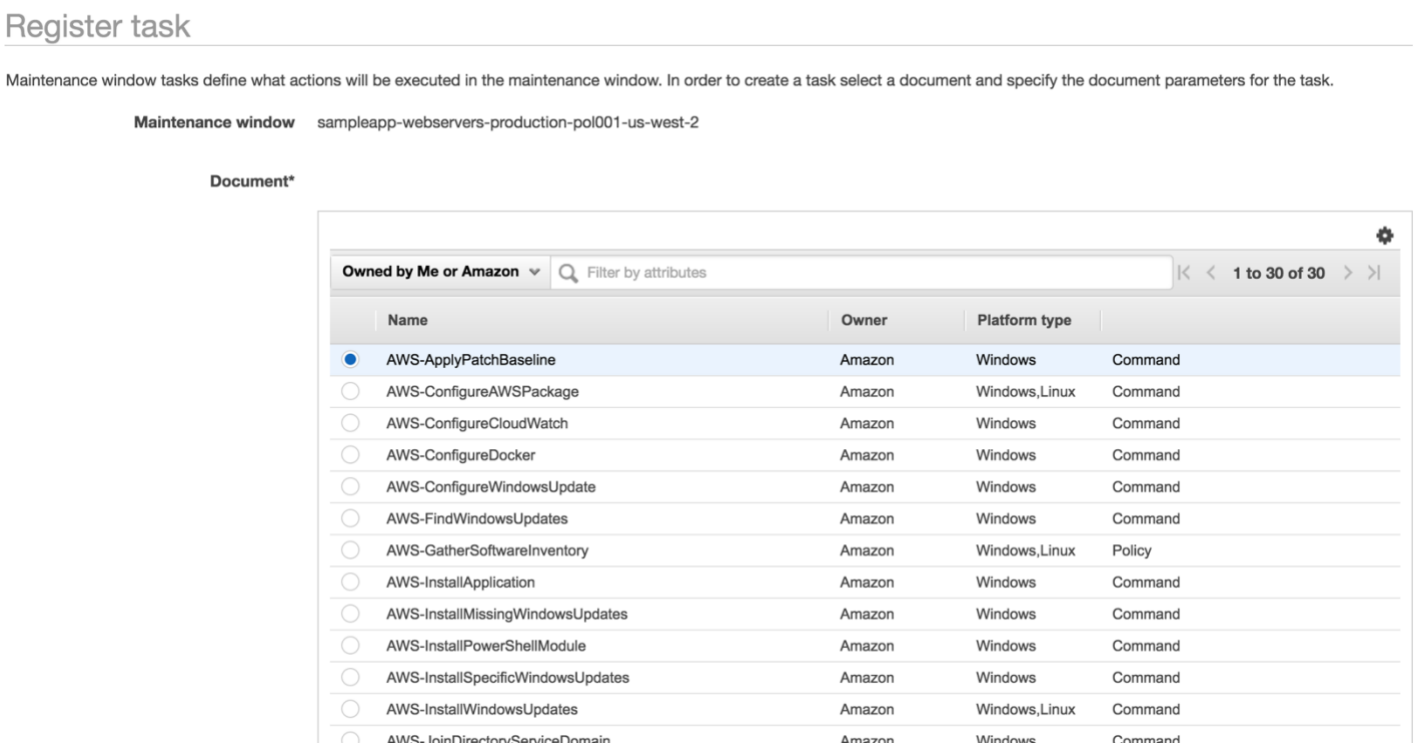
Registering Tasks

To register a task:

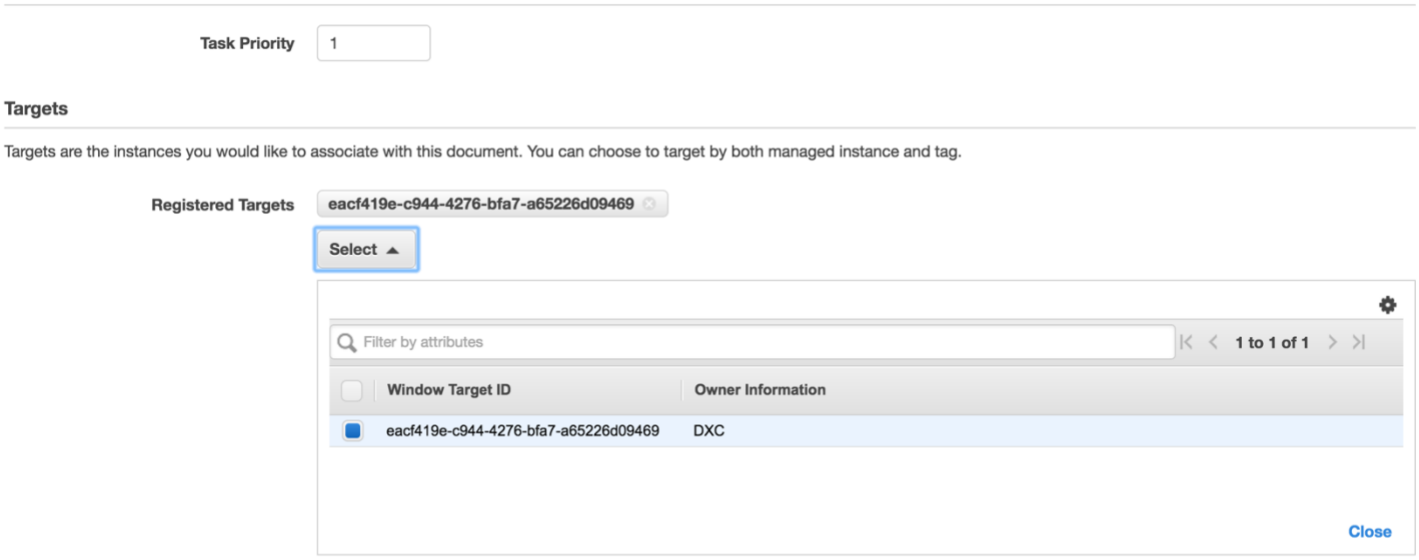
1. Make sure that your maintenance window is selected.
2. Under **Actions**, select **Register task**.



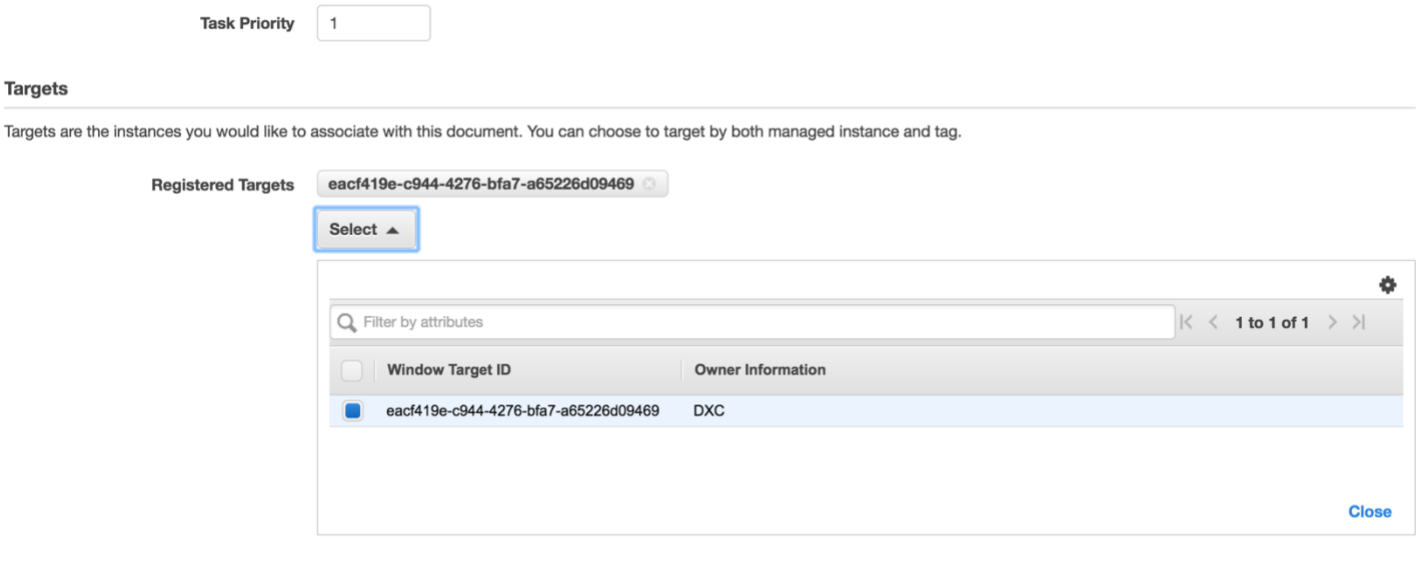
3. Select **AWS-ApplyPatchBaseline**.



4. Make sure that your target is selected and then scroll down and click **Select**.



5. Ensure that the owner you specified before is DXC.
6. Scroll down further and ensure the following:
   - Operation is *install*
   - Role is the role you created or found above, which is arn:aws:iam::284265742084:role/WindowsPatchingRole in the example
   - Indicate how many instances the patching should execute on concurrently
   - Indicate the number of errors that should cause the maintenance window to stop
7. Click **Register task**.



Patching Linux Instances

Using Linux Patch Manager

You can use Patch Manager to create a patch baseline for Linux instances, organize instances into patch groups, and scheduling a maintenance window.

This example illustrates how to create a patch baseline for Linux instances. The baseline will be run under a maintenance window. The maintenance window sets the schedule for applying the patches (baseline) on the specified instances in the patch group. You configure the patch group on the instance by setting the **Patch Group** tag. **Note:** At the time of the release of this document, the SSM agent installed on Linux instances must be at a minimum version of 2.0.834.0.

For specific details on working with Patch Manager for Linux instances, refer to the AWS documentation: http://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-working.html.

Patch Baseline

The patch baseline is used to specify the following criteria about the patching to apply to instances:

- Classification (patch type)
- Severity
- Auto approve delay
- Approved (whitelisted) patches
- Rejected (blacklisted) patches

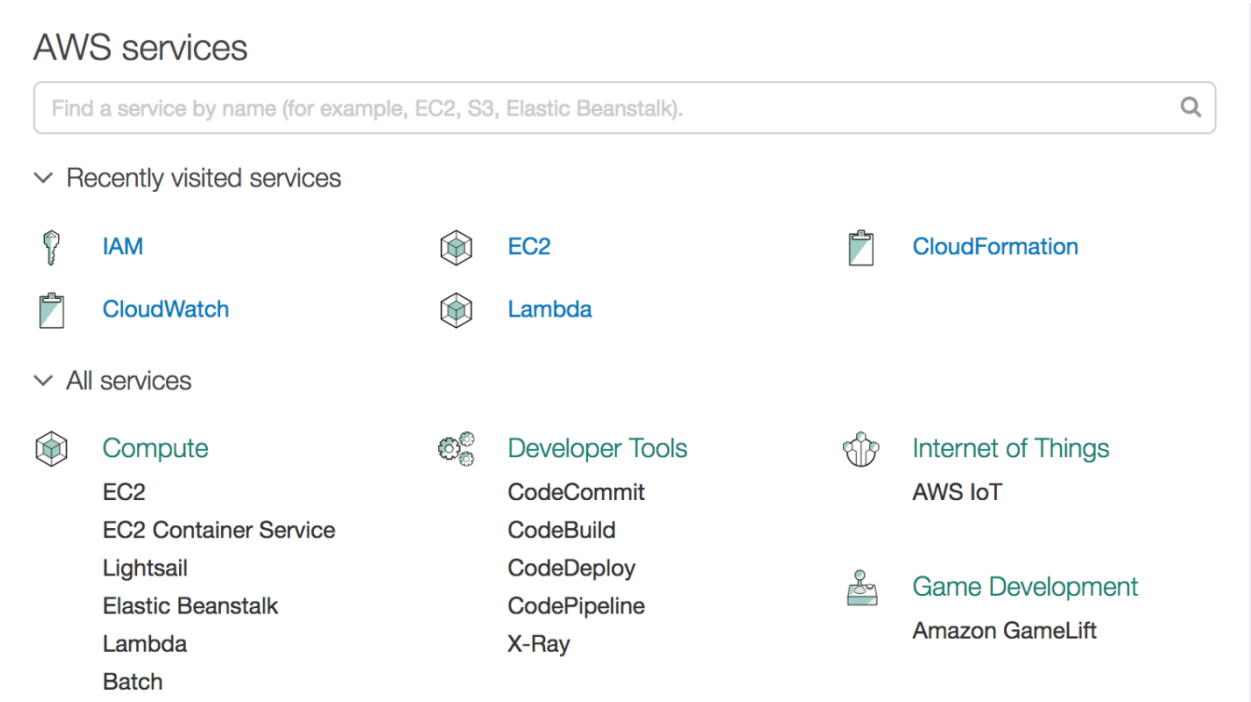
You can search for an existing patch baseline that meets your needs or create a patch baseline to apply to your instances.

Inspecting current Patch Baselines

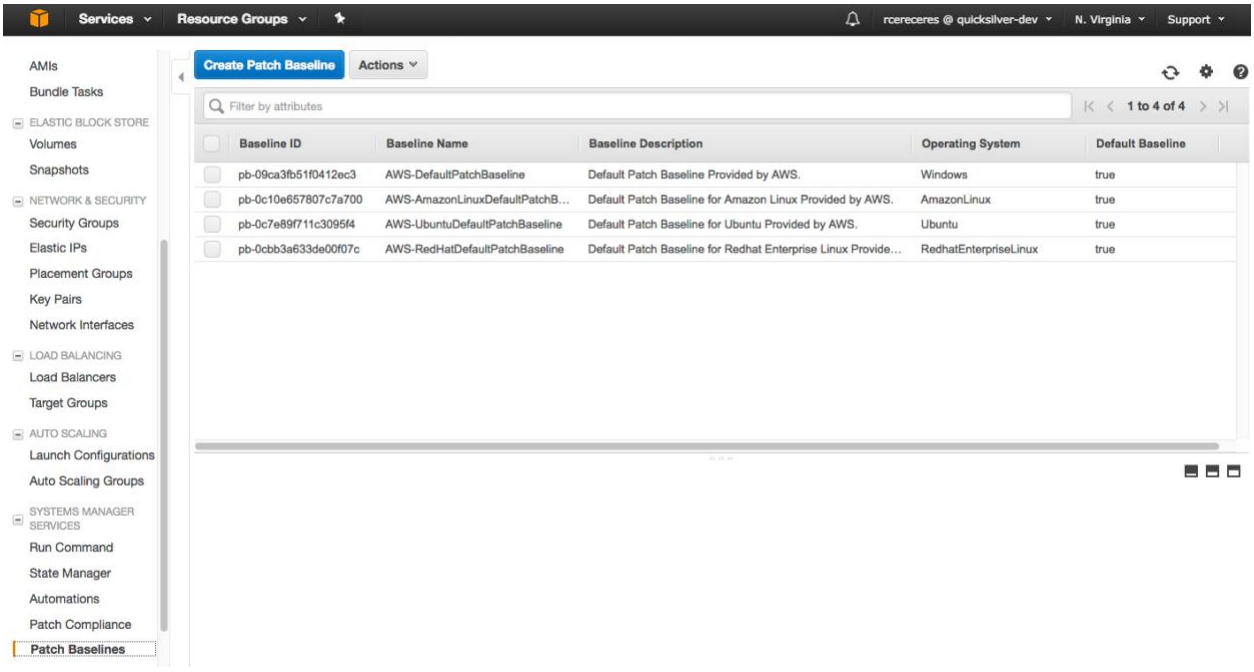
If you are a creating a new patch baseline, you can skip this section. For more information, see the "Creating a Patch Baseline" section.

To inspect current patch baselines:

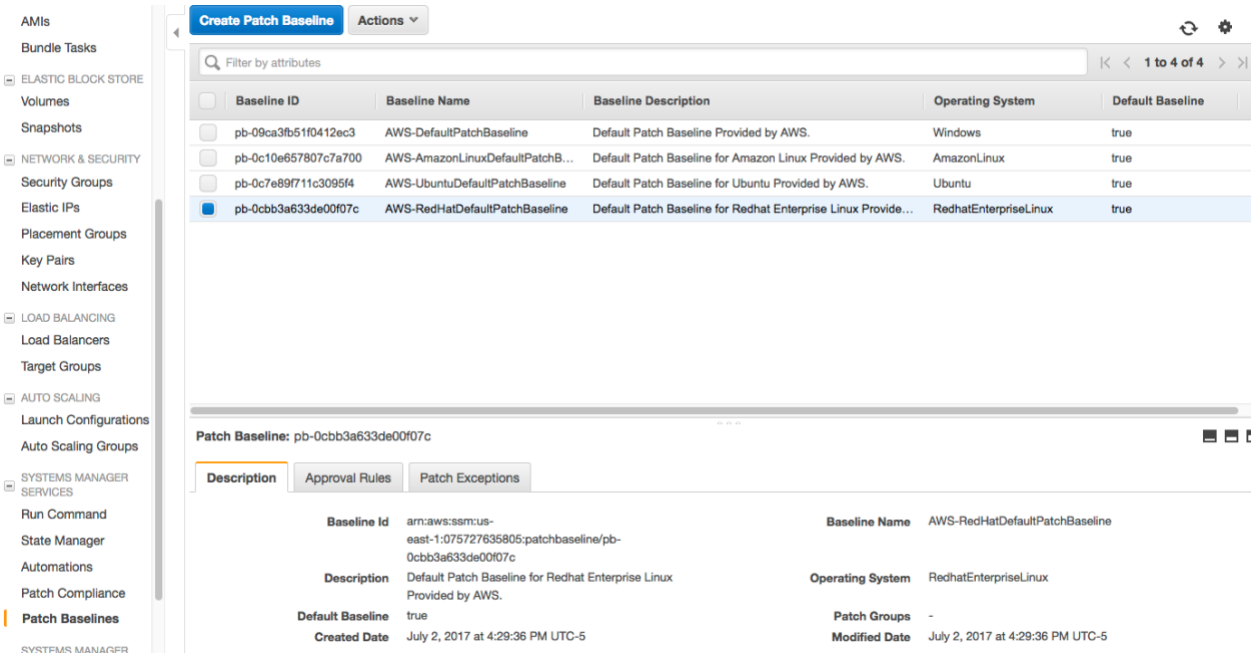
1. On the AWS services page, click **EC2**.



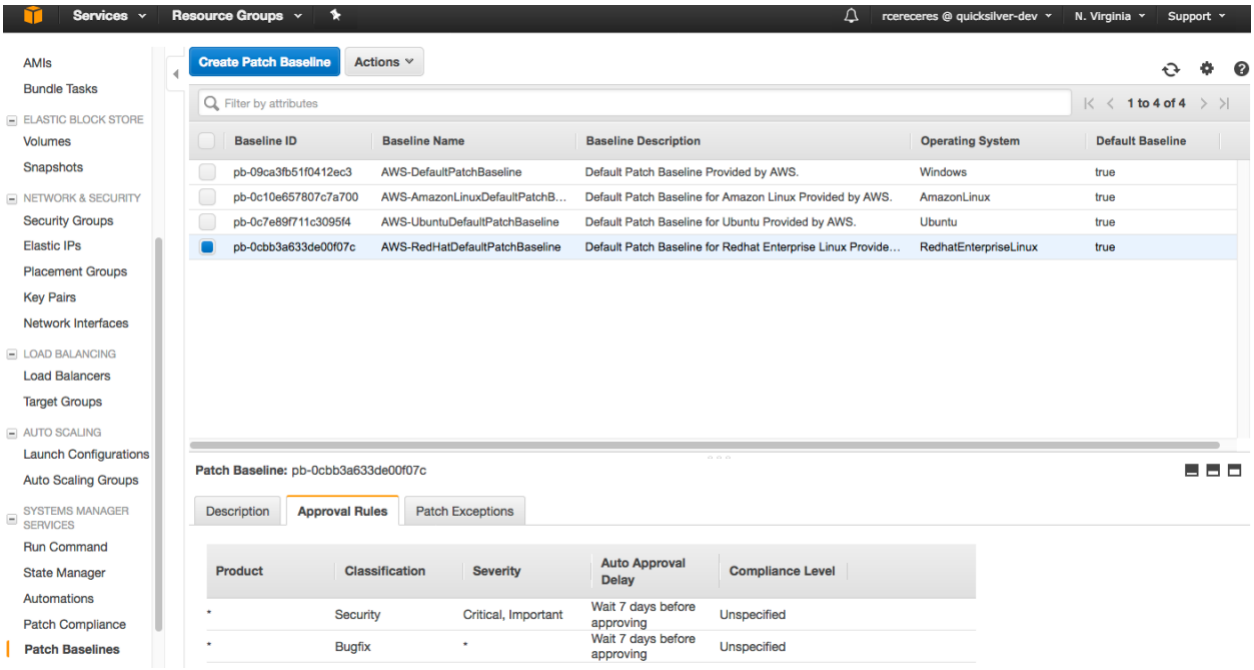
2. Under **SYSTEMS MANAGER SERVICES** on the left nav column, click **Patch Baselines**.



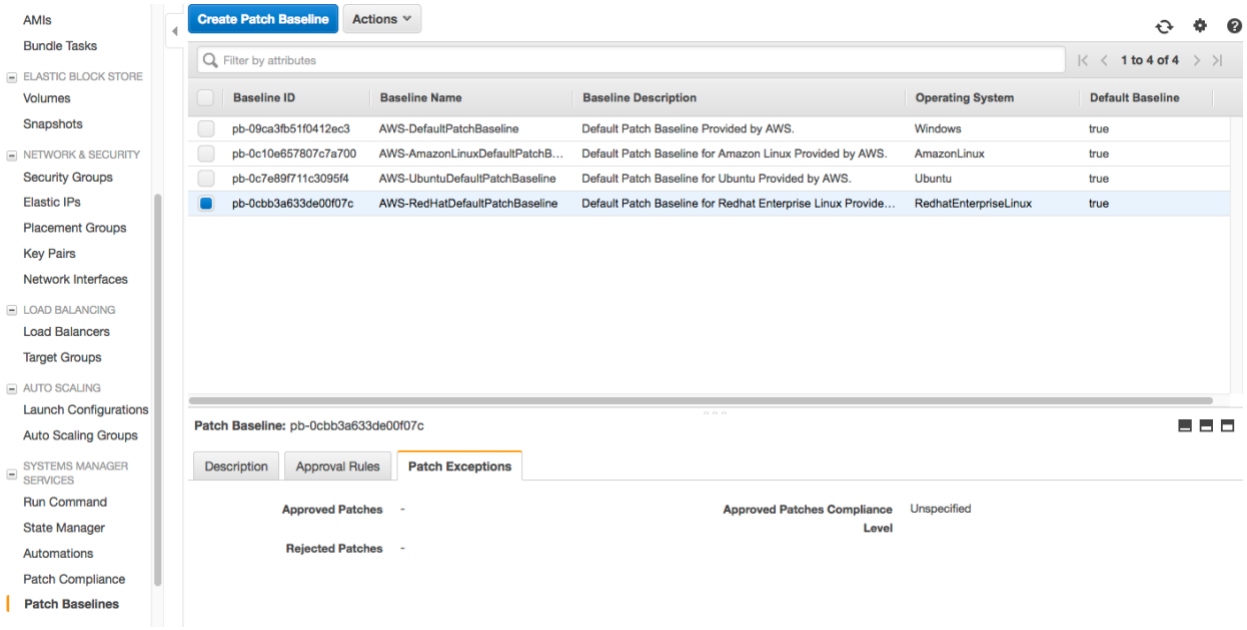
3. Select each patch baseline to view the **Description** tab, **Approval Rules** tab, and **Patch Exceptions** (whitelist and blacklist).



The default RedHat example baseline shown below applies critical and security patches marked with a severity of **Critical** and I**mportant**. The patches are applied 7 days after they are released. Additionally, all bug fixes are also applied 7 days after they are released.



This baseline does not have approved (whitelist) patches nor does it have rejected (blacklist) patches.



Notes:

You can filter the patch baseline list by **Name Prefix**, **Operating System** or **Owner**, and it is recommended that you name a patch baseline to match the patch type to make it easier to identify which baselines apply to each patch type.

Search for applicable patch baselines by executing the above procedure for each patch baseline until you find an existing one that works. You can also create your own patch baseline.

Creating a Patch Baseline

Use a naming convention for patch baselines. For this example the convention will be:

<environment>-<os>-<region>

The example shows how to create a baseline for the development RHEL 6.7 servers in us-west-2.

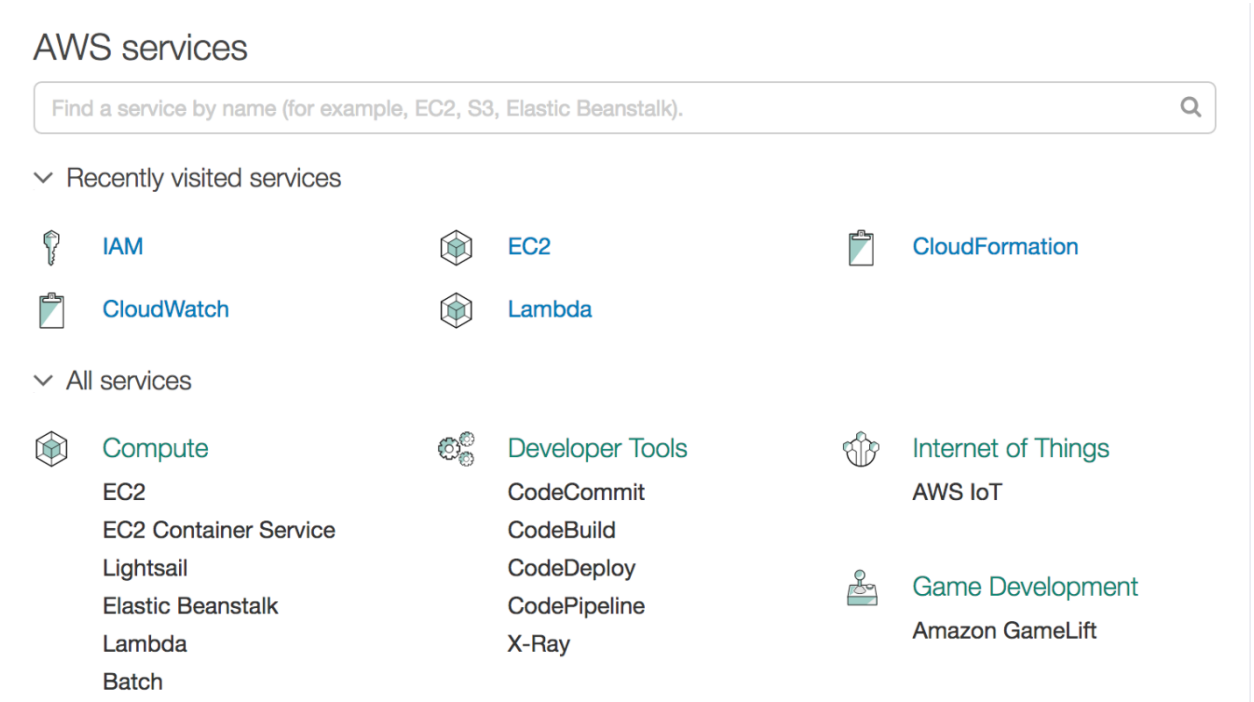
dev-rhel6-patch-us-west-2

The patch baseline has the following requirements for the servers, which are RHEL 6.7 instances:

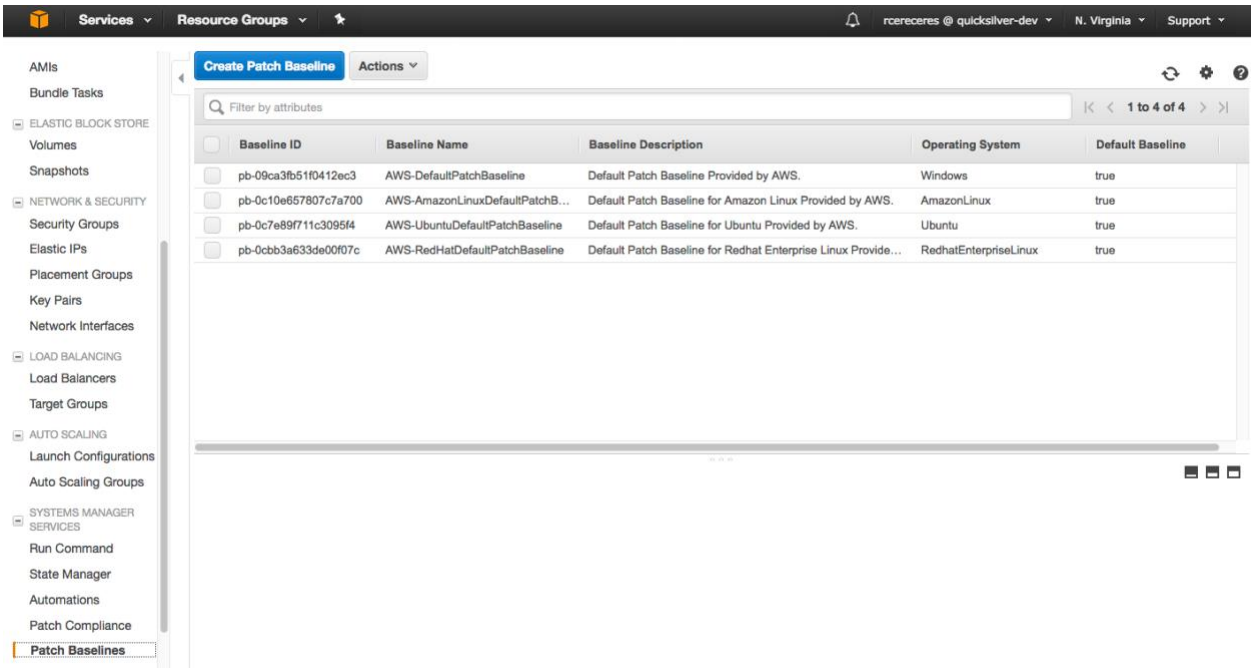
- Apply Security updates of severity Critical 4 days after the release.
- Apply Bugfix updates of level Important 7 days after the release.
- Do not apply patch on any python-related package (blacklist).

To create a patch baseline:

1. On the AWS services page, click **EC2**.



2. Under **SYSTEMS MANAGER SERVICES** on the left nav column click **Patch Baselines**.



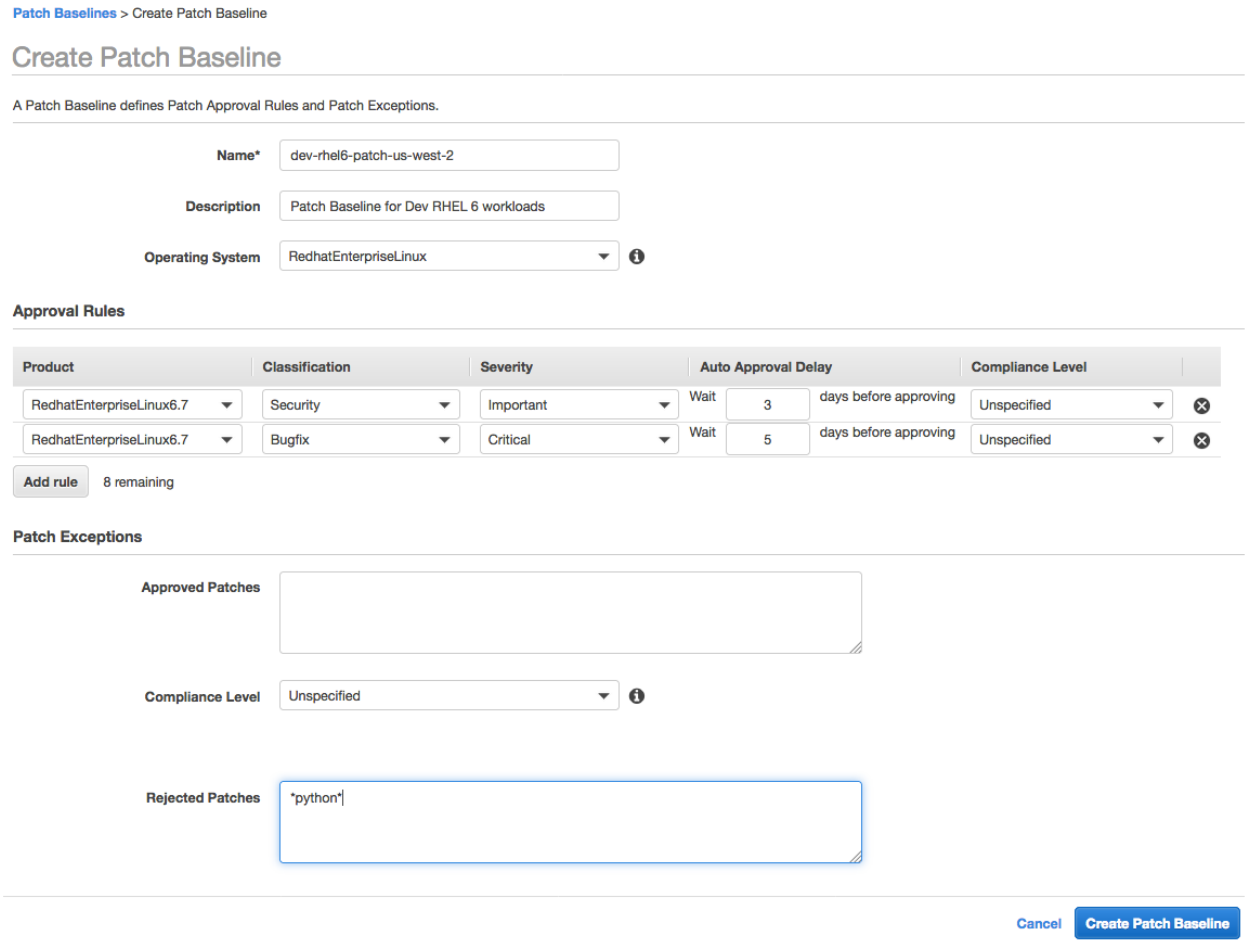
3. Click **Create Patch Baseline**.
4. On the **Create Patch Baseline** page, fill in the following information:

Field	Value
Name	Name of the patch baseline, for this example: **dev-rhel6-patch-us-west-2/**
Description	Type a description for the patch baseline.
Operating System	Select **RedhatEnterpriseLinux** from the dropdown list.

5. Under **Approval Rules**, add the following two approval rules:

Product	Classification	Severity	Auto Approval Delay	Compliance Level
RedhatEnterpriseLinux6.7	Security	Important	3	Unspecified
RedhatEnterpriseLinux6.7	Bugfix	Critical	5	Unspecified

6. In the **Patch Exceptions Section**, under **Reject Patches**, type *python*.
7. Click **Create Patch Baseline**.



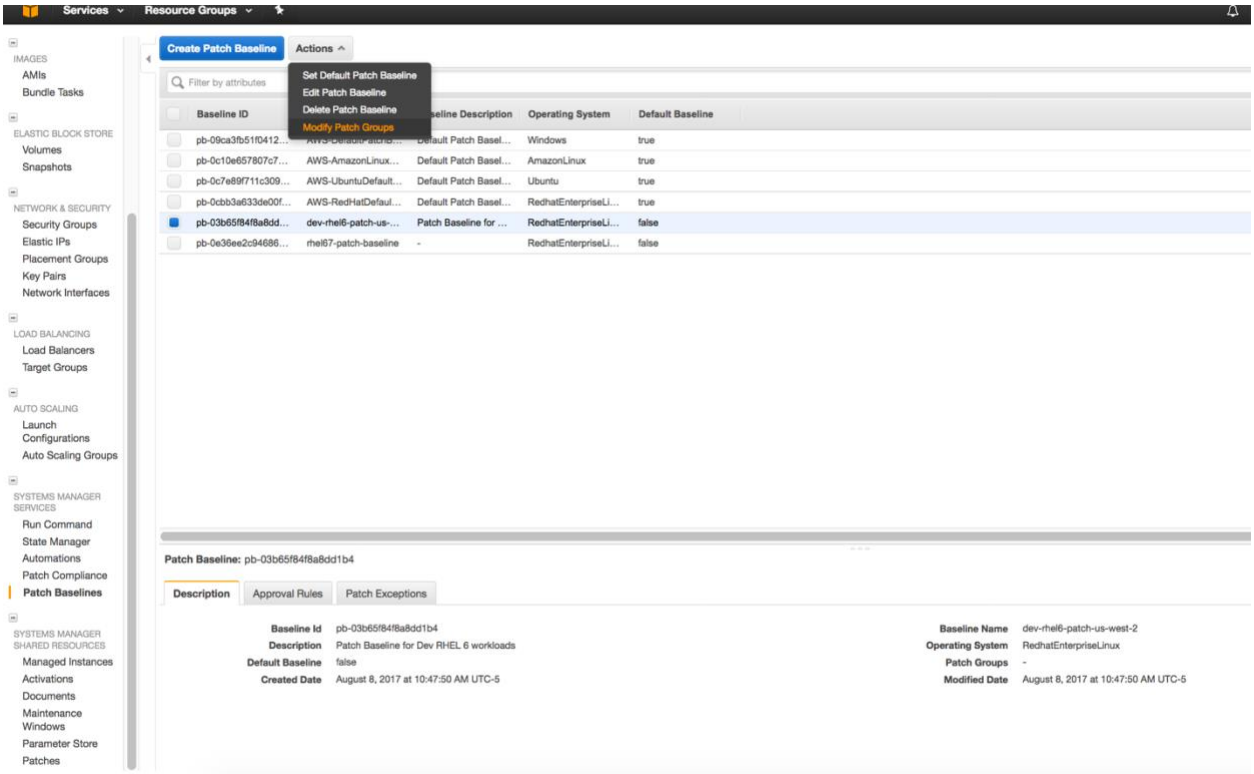
Note: When you create patch baselines for Amazon Linux and RHEL, if you specify **Approved Patches**, be aware that Systems Manager supports Bugzilla ID, CVE ID, Advisory ID, and package-name wildcards. If you specify **Rejected Patches**, Systems Manager only supports package-name wildcards.

Attaching a Patch Group to the Patch Baseline

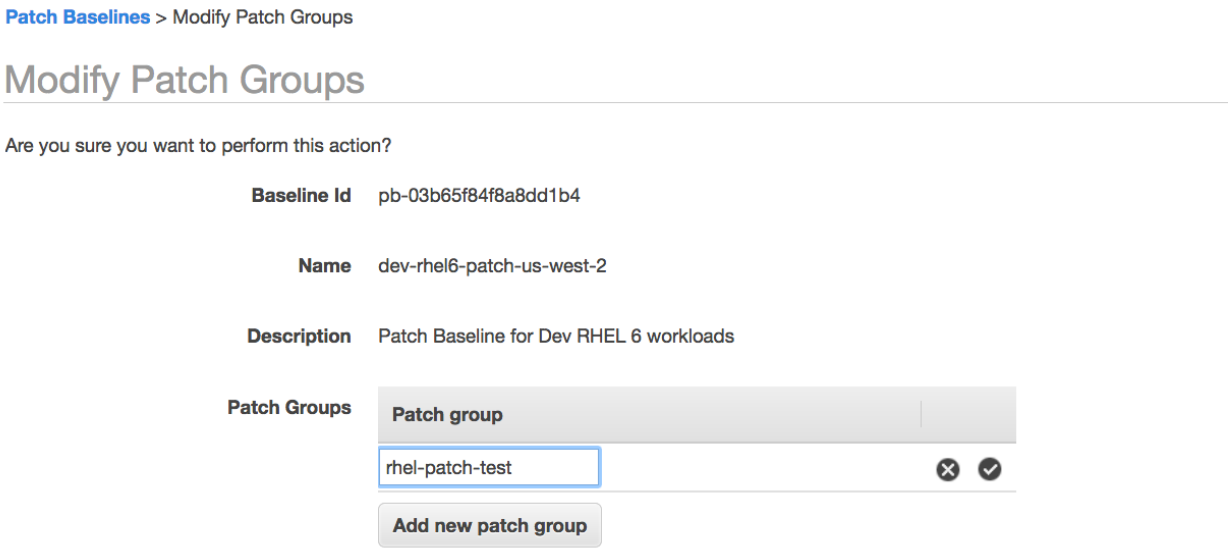
After you have identified or created the correct patch baseline, you need to associate it with a Patch Group.

To attach a patch group to the patch baseline:

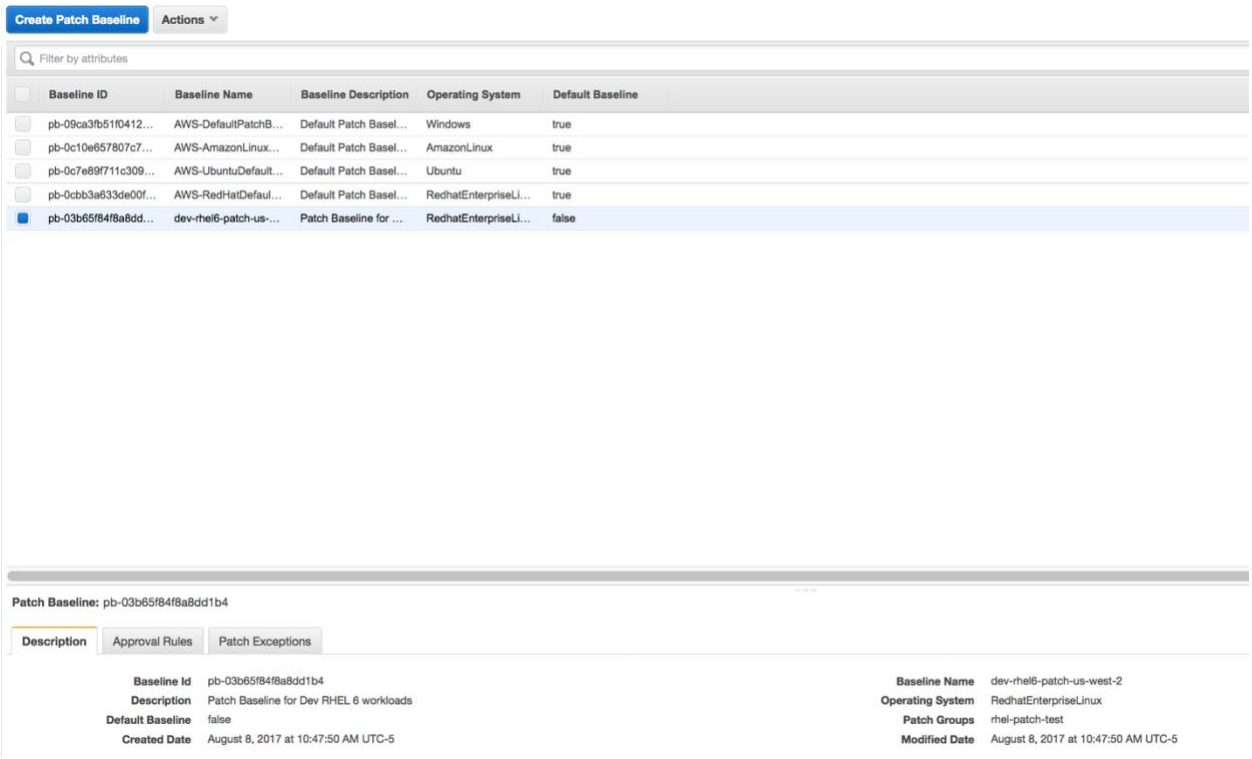
1. In EC2, under **SYSTEMS MANAGER SERVICES** on the left nav column click **Patch Baselines**.
2. Select the patch baseline you have created.
3. Click **Actions** and select **Modify Patch Groups**.



4. If you do not see a patch group text box, click **Add new patch group**.
5. In the text box, type **rhel-patch-test.** Instances that have a tag named **Patch Group** that has this value, will have this patch baseline applied to them.



6. Click the checkbox next to the patch group then click **Close**.
7. On the EC2 page, click **Patch Baselines**.
8. Click refresh (the up/down arrow) and then select your patch policy and ensure that the **Patch Groups** entry under the **Description** tab shows your patch group.



Tagging Instances

Any instances that are started up as a result of creating a stack from the **simple-workload-linux.json** CloudFormation template should have the **Patch Group** tag/value applied if a value was set for the **Patch Group** variable on the template.

The example below describes how to manually add the tag to an instance in EC2.

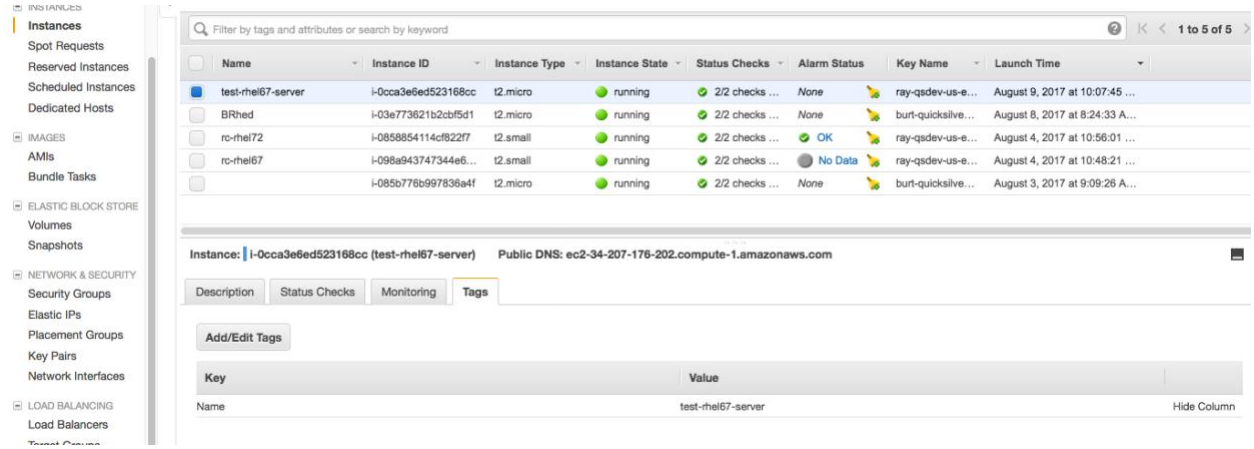
Use the **Patch Group** tag for Linux tagging. The tag value is used to pair the instance with the correct patch baseline in a maintenance window.

Note: It is assumed that the instance(s) have the SSM agent installed and are active, which is required to run commands on the machine.

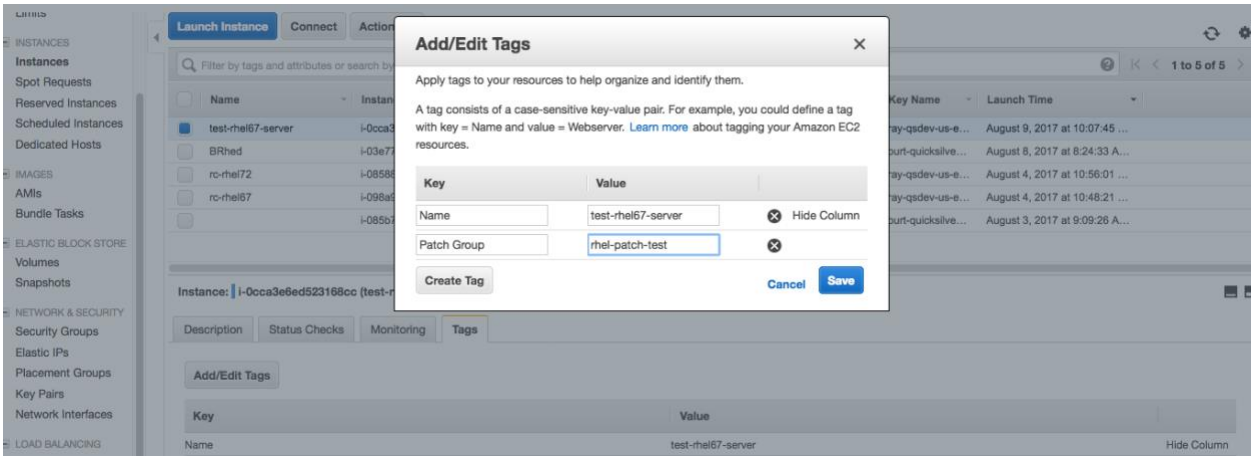
Note: You can apply only one patch baseline to a Windows instance.

To tag an instance:

1. In EC2, click **Running Instances**, locate your instance, and select it.
2. Click the **Tags** tab below the instance list.

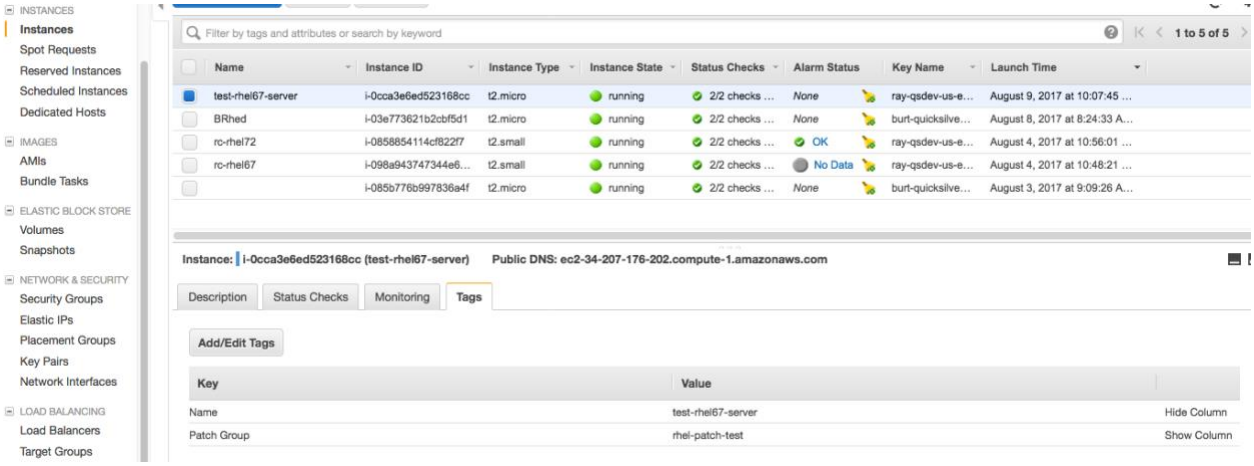


3. Click **Add/Edit Tags**.
4. Click **Create Tag** and add a tag named **Patch Group** with a value of **rhel-test-patch**.



5. Click **Save**.

The instance tags should look like this:

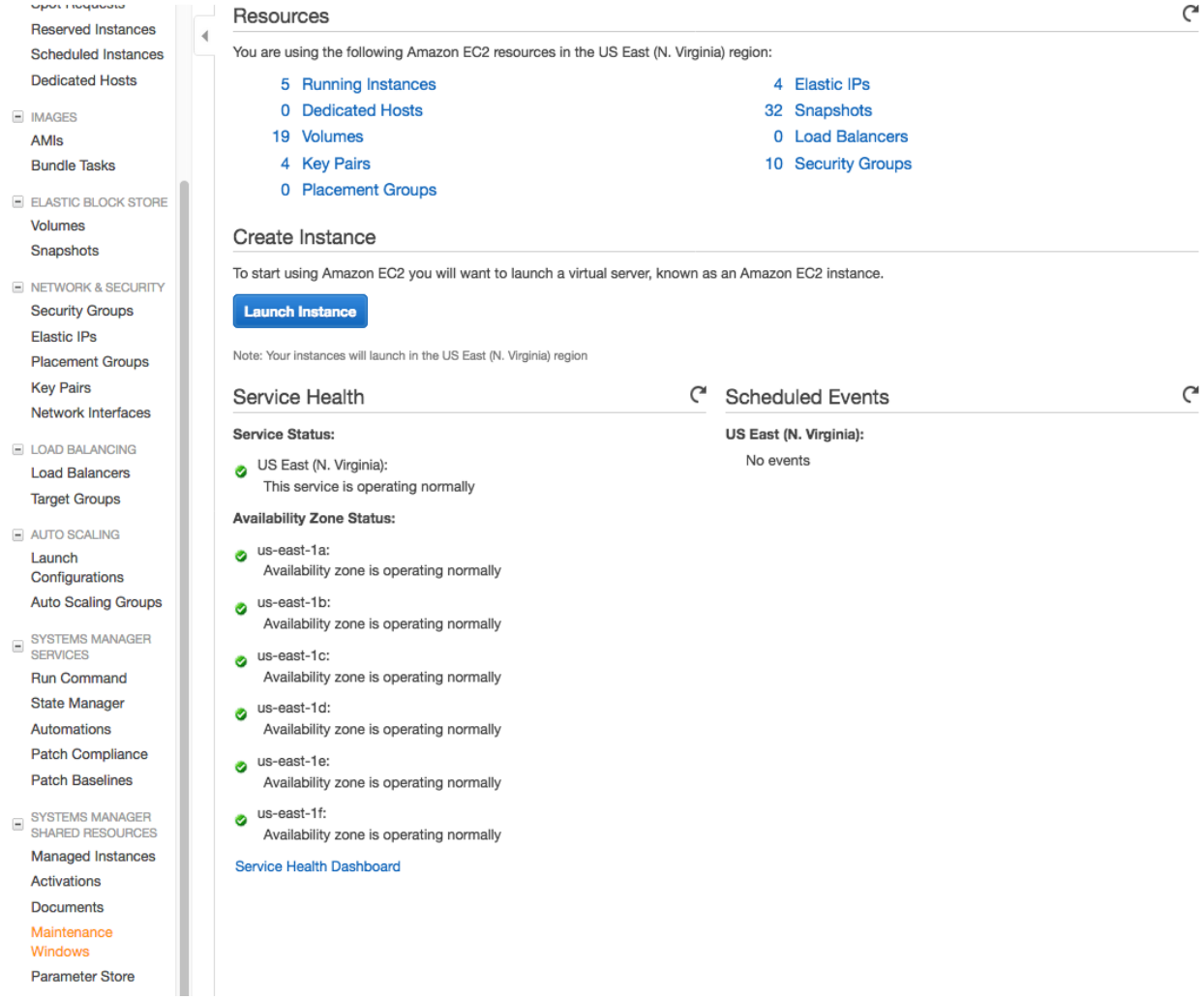


Creating the Maintenance Window

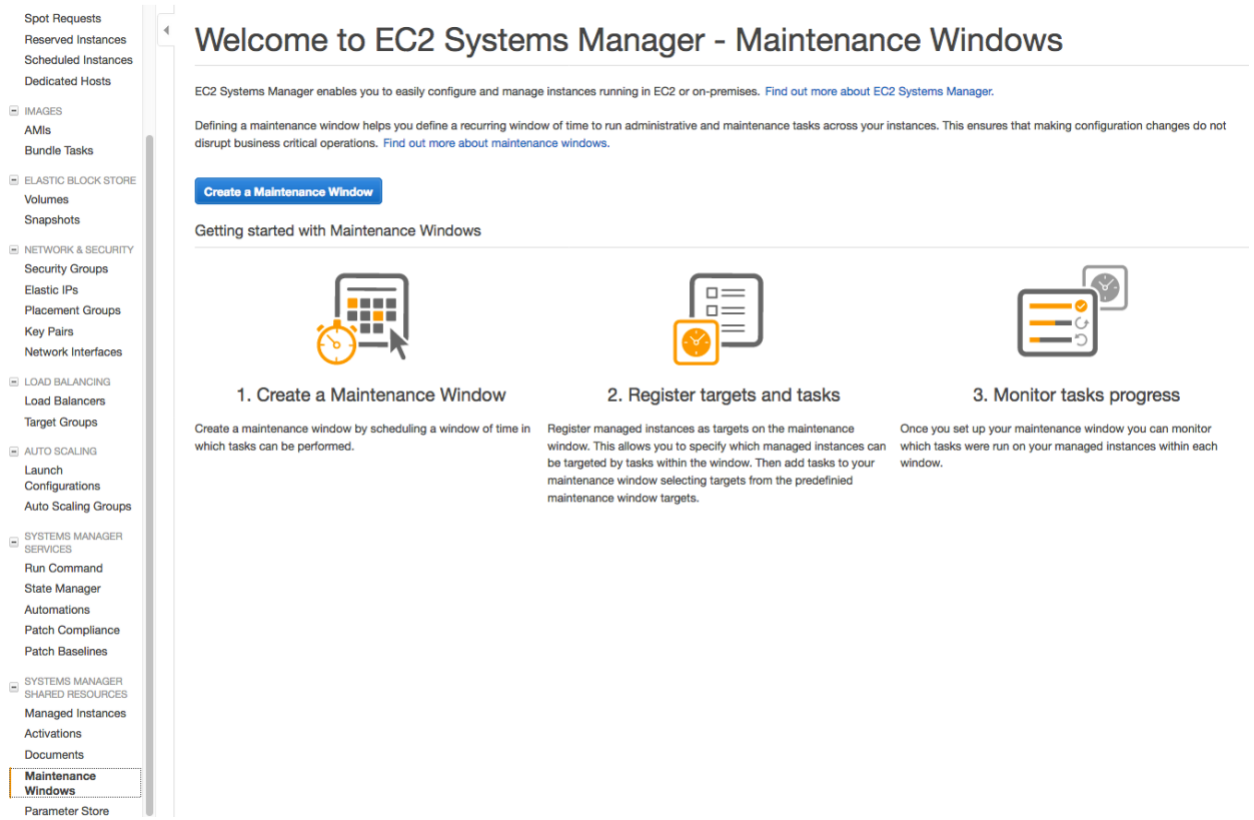
A maintenance window will automate the installation of patches, ensuring that servers patch levels are maintained at a level required by your organization.

To create a maintenance window:

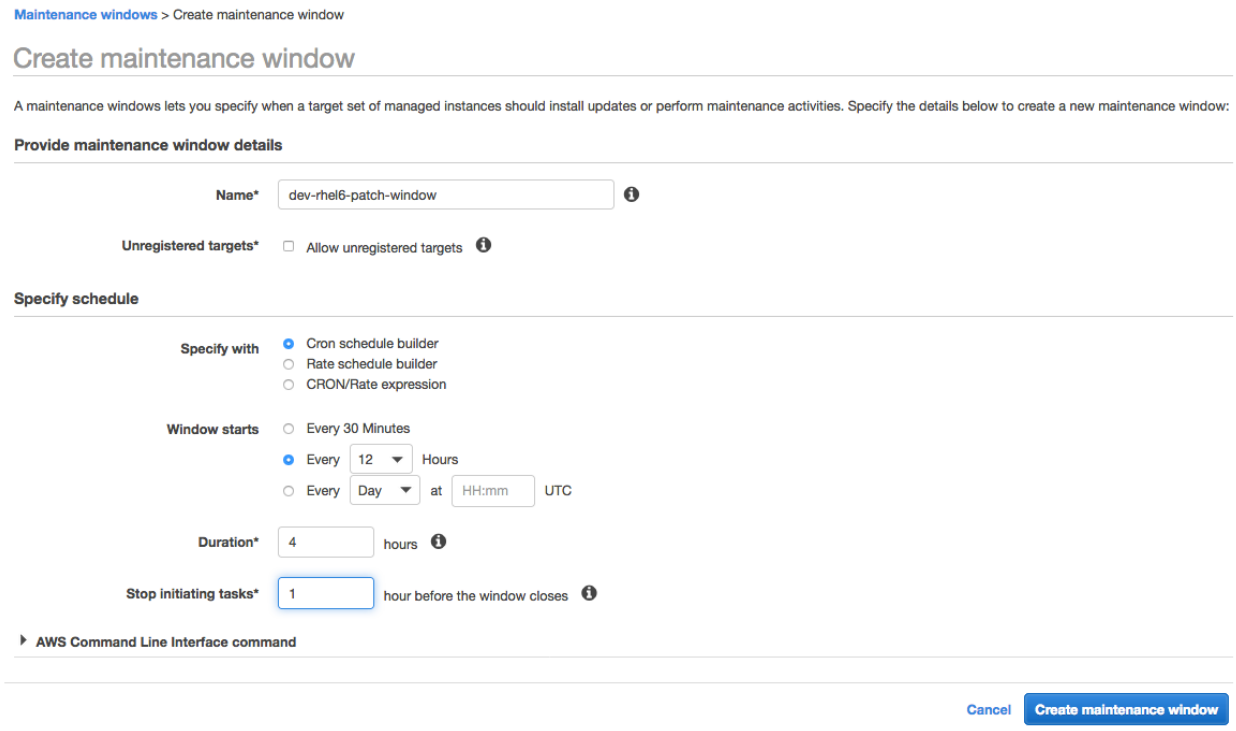
1. In EC2, under **SYSTEMS MANAGER SHARED RESOURCES** on the left nav column, click **Maintenance Windows**.



2. Click **Create maintenance window**.



3. For **Name**, enter **dev-rhel6-patch-window**.



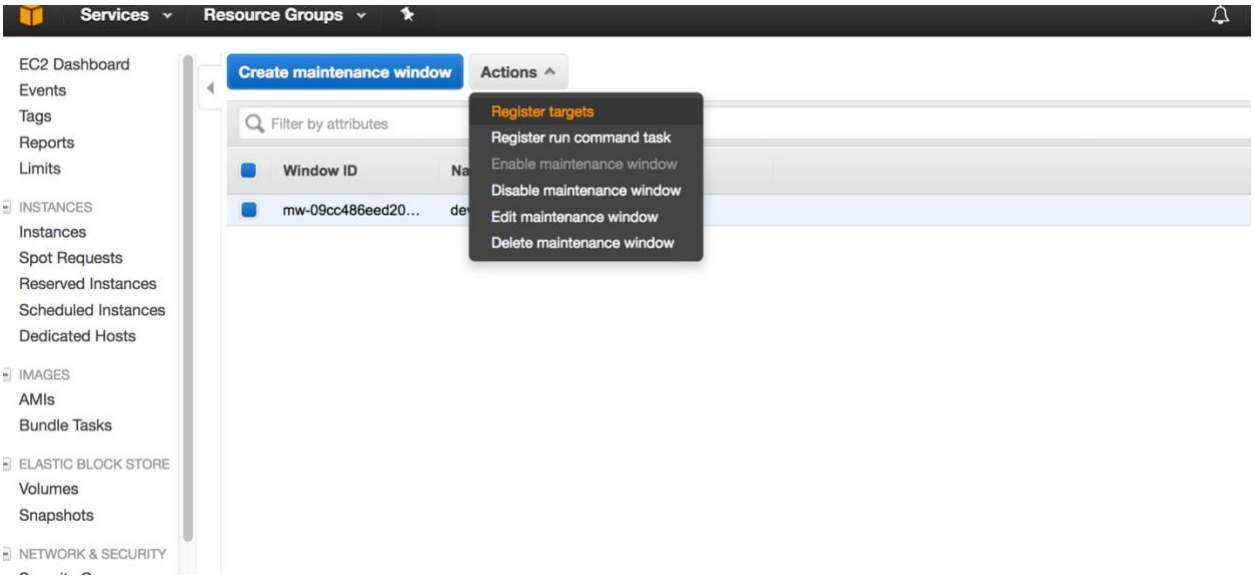
4. For **Window starts**, specify a time for how often the window should start.
5. Specify values for **Duration** and **Stop initiating tasks**.
6. Click **Create maintenance window**.

Registering Targets

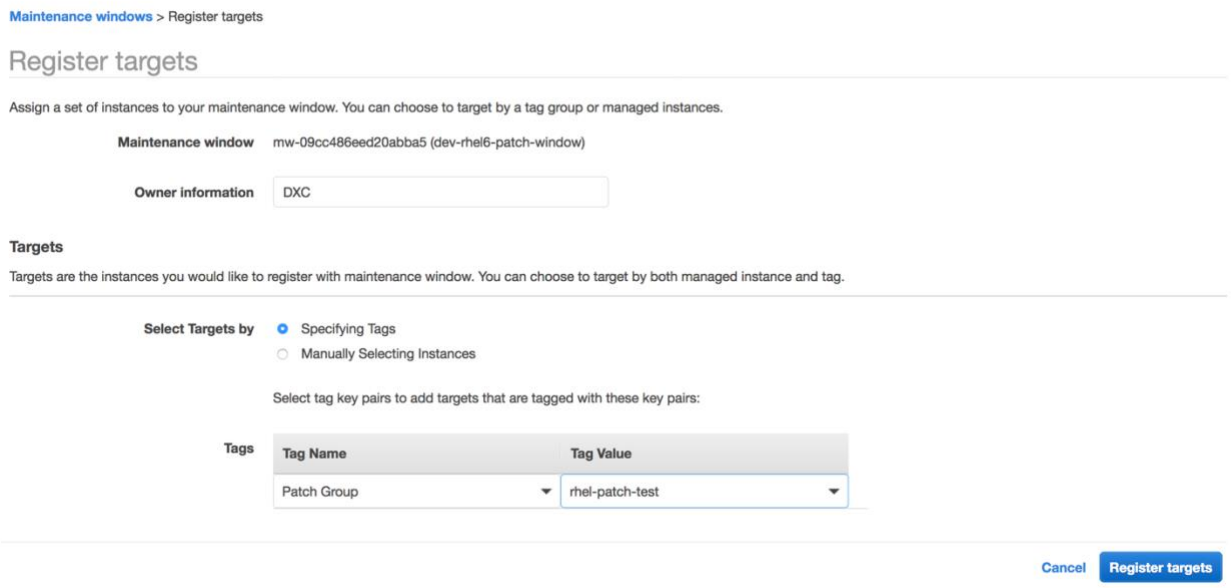
After you have created a maintenance window, you need to register the targets for the maintenance window. The targets are the instances associated with the **Patch Group** tag that have the value **rhel-patch-test**, which is the value associated with your patch baseline.

To register targets:

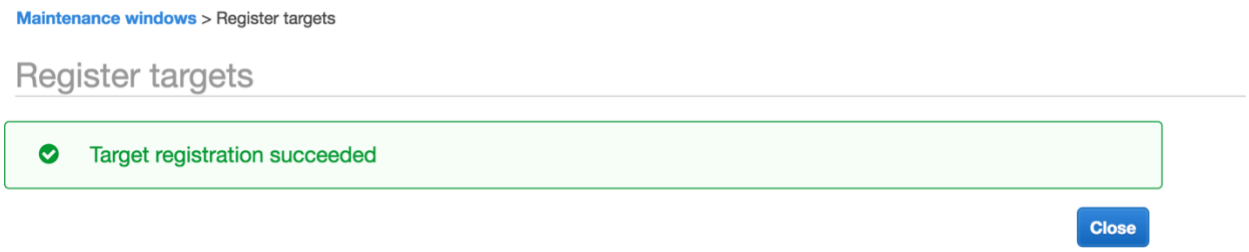
1. In EC2, under **SYSTEMS MANAGER SHARED RESOURCES** on the left nav column, click **Maintenance Windows**.
2. Select your maintenance window.
3. Click **Actions** and select **Register targets**.



4. For **Owner information**, type **DXC**.



5. For **Select targets by**, click **Specifying tags**.
6. For **Tag Filters**, set the **Tag Name** to **Patch Group** and the **Tag Value** to **rhel-patch-test.**
7. Click **Register targets**.
8. Click **Close**.

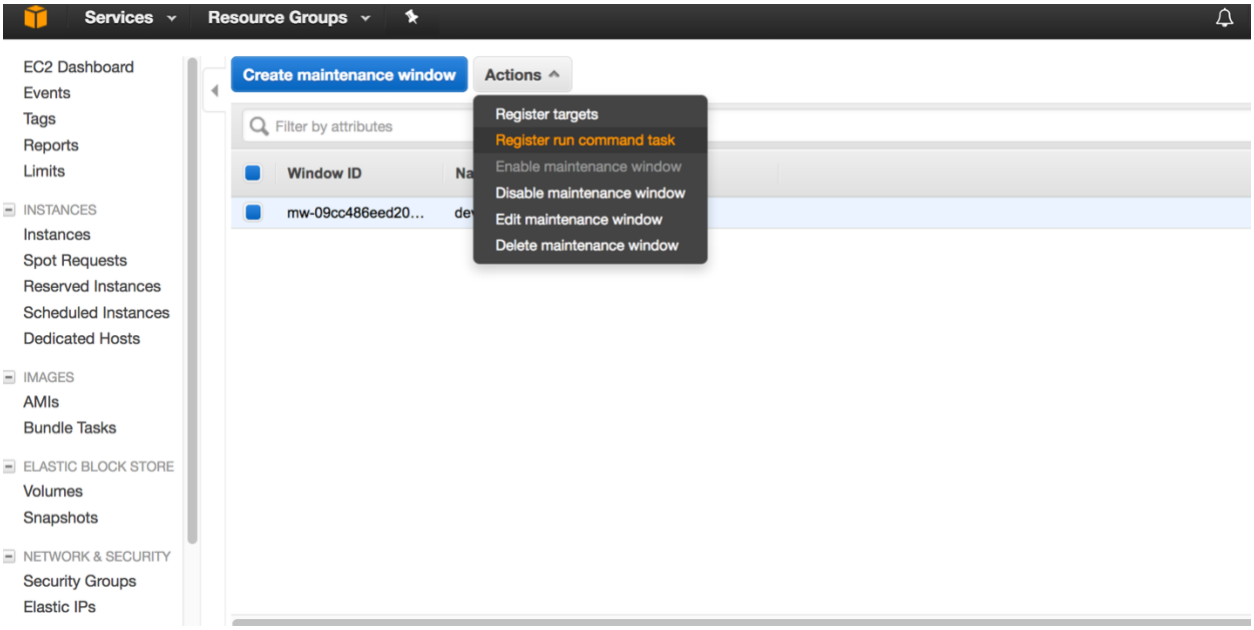


Registering Tasks

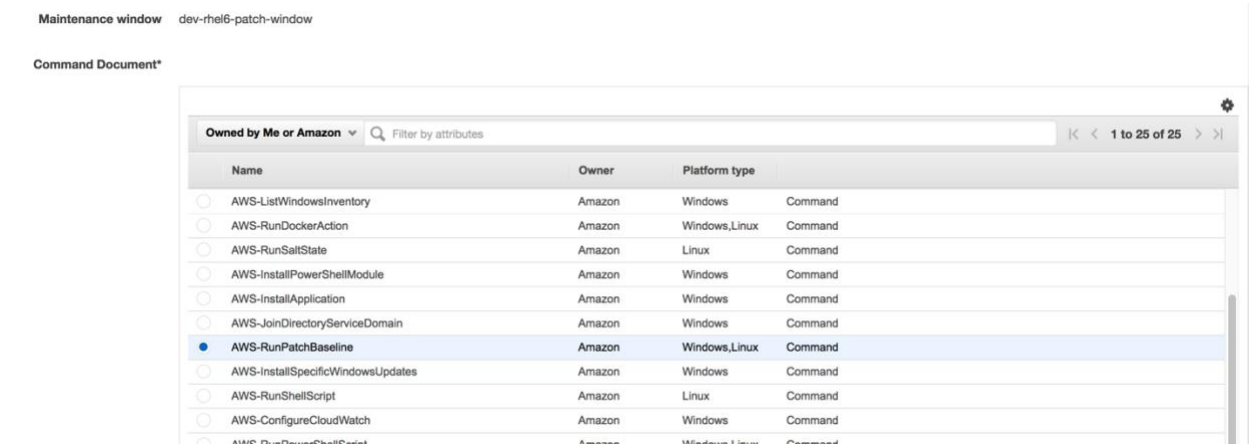
After registering your targets, you need to register your tasks.

To register your tasks:

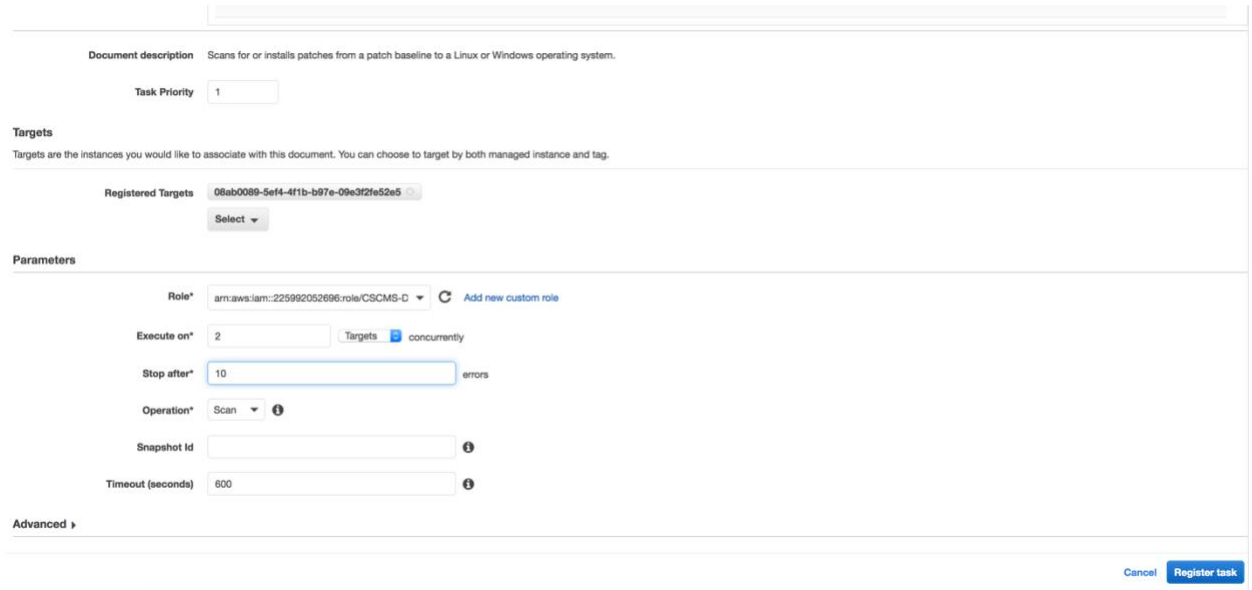
1. In EC2, under **SYSTEMS MANAGER SHARED RESOURCES** on the left nav column, click **Maintenance Windows**.
2. Select your maintenance window.
3. Under **Actions**, select **Register run command task**.



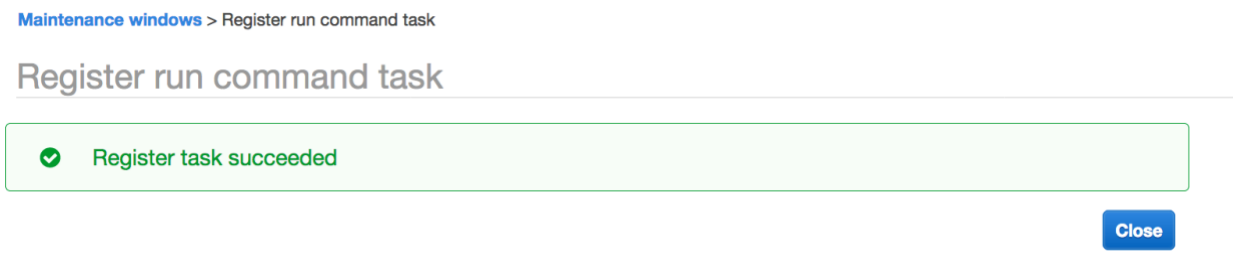
4. Select **AWS-RunPatchBaseline**.



5. Scroll down to the **Targets** section and verify that your target is selected. If no target, has been selected, click **Select**, and select a target.



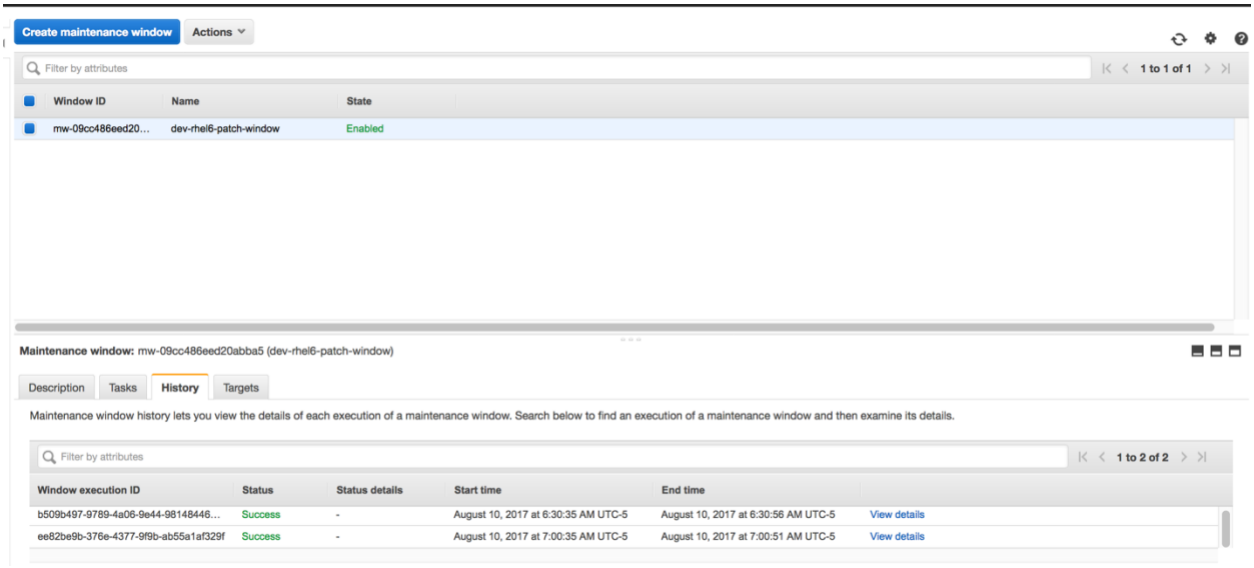
6. Verify that the owner you specified is **DXC**.
7. Scroll down to the **Parameters** section and ensure the following values are set:
   - **Role** should be the DefaultInstanceRole created by the Master template, which is **arn:aws:iam::225992052696:role/CSCMS-Default-Instance-Role** in the example.
   - **Operation**: select **Install**.
   - For **Execute on**, specify how many instances the patching should execute on concurrently.
   - For **Stop after,** specify the number of errors that should cause the maintenance window to stop.
8. Click **Register task**.



9. After the registration has successfully completed, click **Close**.

Viewing the Maintenance Window Results

To view the results of the Maintenance Window executions, select the maintenance window and click the **History** tab. This will display the runs. To view additional details, click **View details**.



Monitoring Instances

12

The following CloudWatch alarms are defined when an instance is created:

Alarm	Description
System Status Failed Alarm	AWS/EC2 system status check failed for 3 consecutive periods of 60 seconds
Status Failed Alarm	AWS/EC2 status check failed for 3 consecutive periods of 60 seconds
Memory Utilization Alarm	Minimum memory utilization over 85% for 2 consecutive periods of 300 seconds
Instance Status Failed Alarms	AWS/EC2 instance status check failed for 3 consecutive periods of 60 seconds
Disk Utilization Alarm	Minimum disk utilization on root volume over 85% utilization for 2 consecutive periods of 300 seconds
CPU Utilization Alarm	Average CPU utilization over 85% for 2 consecutive periods of 300 seconds

EC2 Alarms are discussed on this page:

http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ec2-metricscollected.html

Some of the metrics above are custom (disk and memory).

Protecting Endpoints

13

The End Point Protection service is provided by CrowdStrike, a SaaS based solution. The agent is preinstalled on the Windows AMIs and requires the Delivery Engineer enter in the CID, which is a unique ID that differentiates one customer from another. For Linux, the agent must be downloaded from the CrowdStrike portal, placed in the customer S3 bucket, and installed during the provisioning of the Linux instance.

Prerequisites

- Customer has been onboarded onto CrowdStrike
- Delivery Engineer has access to CrowdStrike portal
- Google Chrome is the only supported browser

Downloading the Linux RPM

To download the RPM for Linux installations:

1. Log into CrowdStrike.
2. Navigate to **Support** and then **Downloads**.
3. Look for the download **RHEL September 2016**.
4. Download the Falcon Host Sensor to your local drive. Currently the name of the RPM is **falcon-sensor-2.0.24-1404.x86_64.rpm**.
5. Copy the RPM to each customer S3 bucket for each region where instances will be provisioned. The location of the file should be:
   dxc.customer.config-<AWS::AccountId>-<AWS::Region>/deploy/externs/falcon-sensor-2.0.24-1404.x86_64.rpm
   For example:**dxc.customer.config-211682634048-us-west-2/deploy/externs/falcon-sensor-2.0.24-1404.x86_64.rpm**. The RPM will be retrieved from this S3 bucket during instance provisioning

Finding the CrowdStrike ID for Windows Instances

For Windows installations, the CrowdStrike ID is required. To find the customer's CrowdStrike ID:

1. Log into CrowdStrike.
2. Navigate to **Support** and then **Downloads**.
3. At the top of the screen, you should see the message, "Your Customer ID with Checksum is 5F983CA84F27459B9B2A025AC483EEB9-C0."
4. Copy this ID for later use when provisioning Windows instances.

Installing the Falcon Host Sensor

You can install the Falcon Host Sensor on Linux or Windows instances.

Linux Instances

Falcon Host Sensor is installed on Linux instances as part of a provisioning process. The agent is copied from the customer's S3 bucket to the local file system and installed. No additional parameters are required.

Windows Instances

Falcon Host Sensor is installed on Windows instances as part of the provisioning process. The Windows executable is already present in the customer's S3 bucket. The executable is copied from the customer's S3 bucket and installed during instance provisioning. The Windows installation requires the customer's CrowdStrike ID as a parameter during installation.

Verifying Sensor Visibility in the Cloud

Verify the newly installed sensor in the Falcon Host UI.

1. To view a list of newly installed sensors in the past 24 hours, go to https://falcon.crowdstrike.com.
2. Navigate to **Hosts**.
3. Filter by Operating System.

The hostname of your newly installed sensor appears on this list within 5 minutes of installation. If you do not see your host listed, read through the Sensor Deployment Guide for your platform to troubleshoot connectivity issues.

At this point, you should have activated your Falcon Host account, deployed a sensor in your organization, and verified that the system can be seen in your Falcon Host UI.

Managing Remote Instances

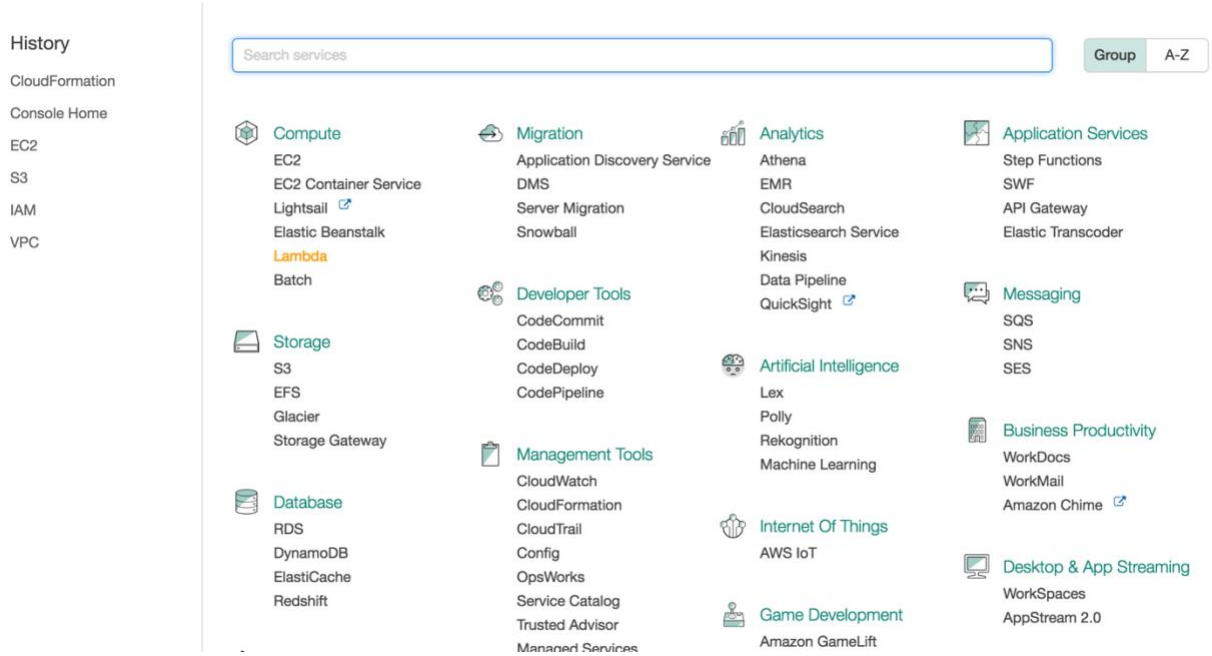
14

This solution allows for the deployment of a Bastion host on a public subnet that can be used to access a managed (target) instance on a public or private subnet for a limited period of time. This solution should only be used if EC2 Run Command is failing to complete the desired task by the System Admin. The solution requires peering and routes between the public subnet and the private subnet of the managed instance. The solution creates the same temporary account on the Bastion host and the managed host. Finally, it creates security groups to allow for client traffic to the Bastion host and Bastion traffic to the managed instance.

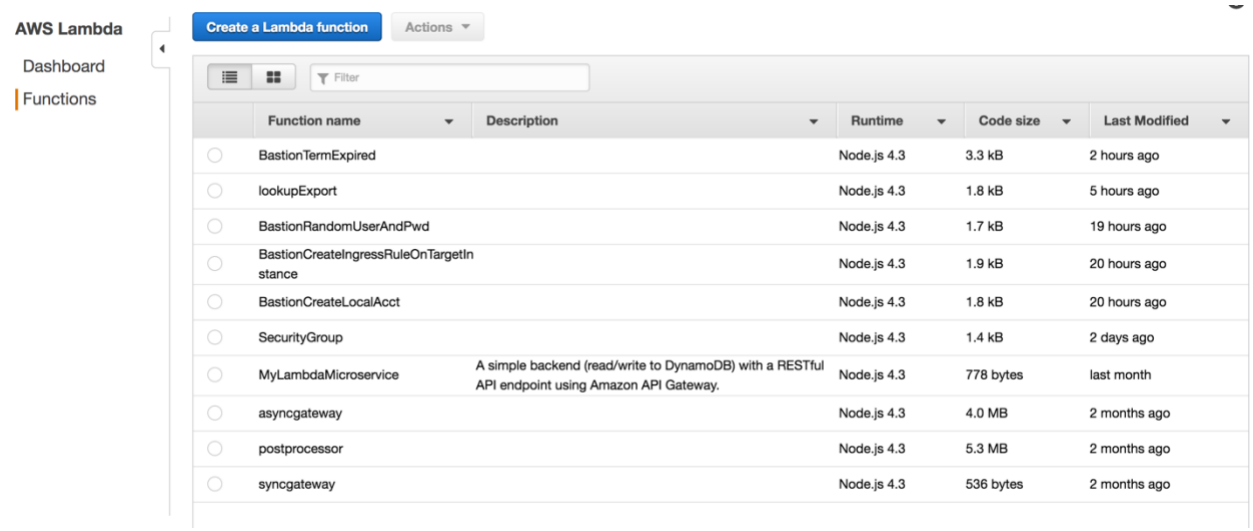
Prerequisites

The Bastion solution depends on the existence of roles and Lambda functions in the region where the Lambda host and the managed instance will reside. To test that these roles and Lambda functions are in place, do the following:

1. In the AWS console, click **Lambda**.



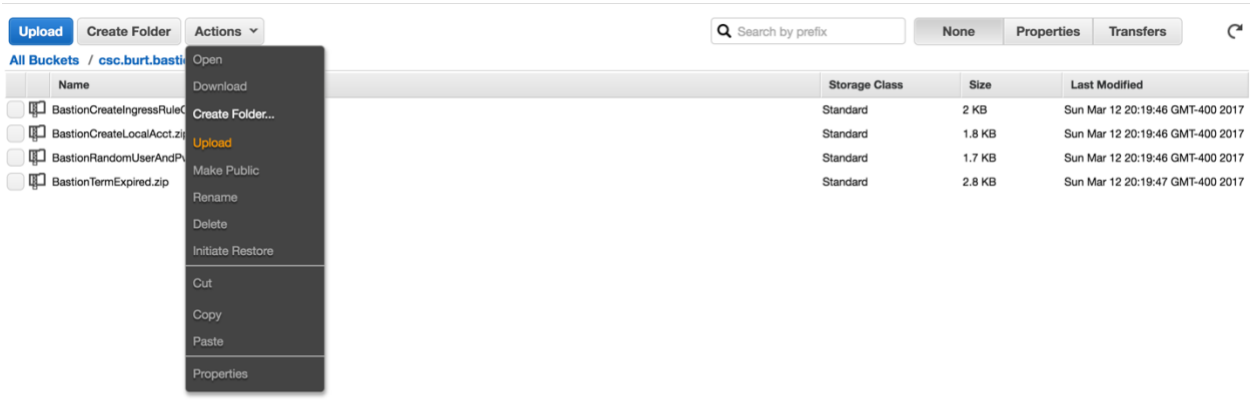
2. List the existing Lambda functions in the region.



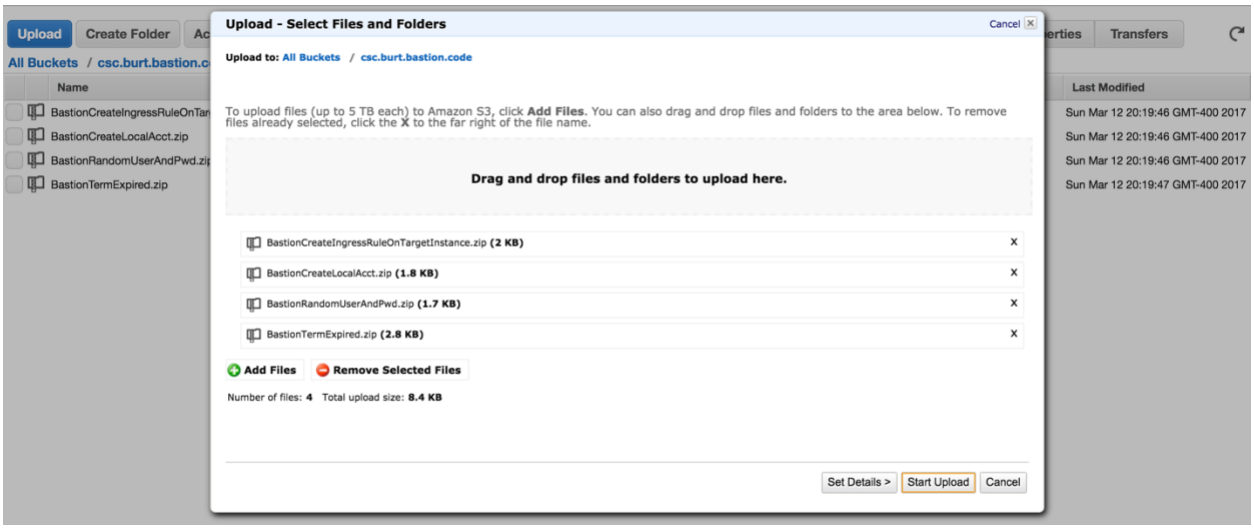
The following functions should be present:

- BastionCreateIngressRuleOnTargetInstance
- BastionRandomUserAndPwd
- BastionCreateLocalAcct
- BastionTermExpired

3. If you do not see these functions, you must create them by running a CloudFormation template. The template assumes that the Lambda functions are in the referenced S3 bucket. This example assumes that the bucket is *dxc.burt.bastion.code* (which is referenced in the yaml below used to create the Lambda functions).
4. Navigate to the S3 bucket.
5. Under **Actions**, click **Upload**.



6. In the **Upload** dialog, select your functions.



Note: These zip files are created from the solution source by running the **ant deploy** command in the *dxcms-cli/bastion-service* directory.

7. After the functions are loaded into S3, run the CloudFormation template https://github.com/dxchcs/dxcms-cli/blob/master/bastion-service/cloudformation/cft_bastion_ondemand_core.yaml to create the following:
   - Lambda functions
   - CloudWatch event for Bastion termination
   - Roles

Note: For information about creating the template under CloudFormation, review the *Running the CloudFormation Template* section.

Assumptions

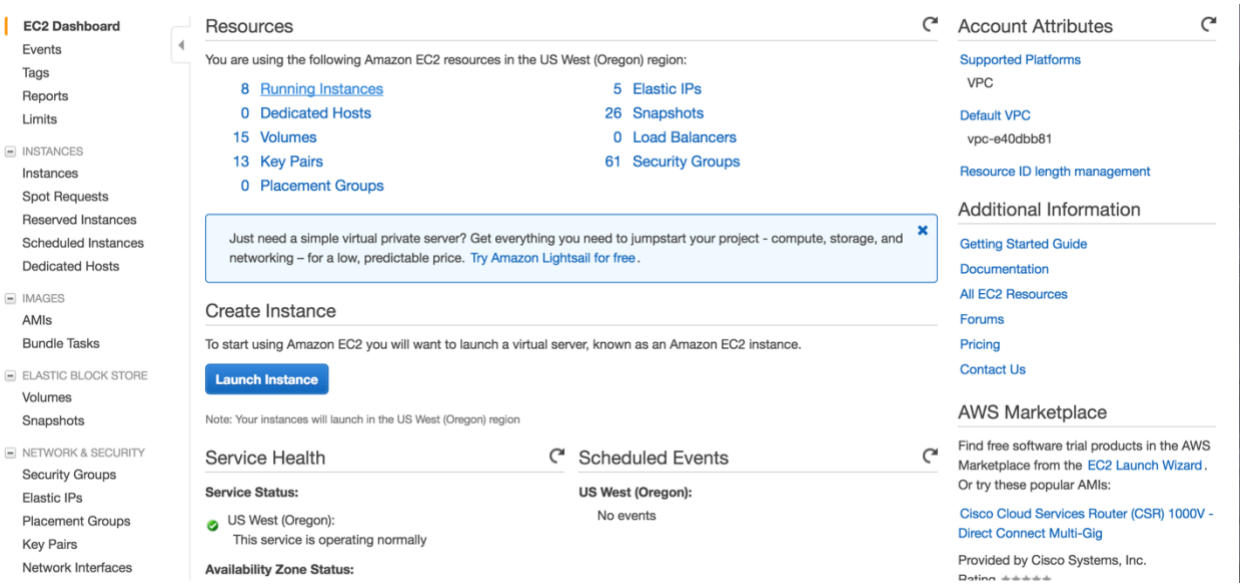
The solution has basic assumptions:

- The managed (target) instance has the ssm agent installed and active, which is required to run commands on the machine.
- If the managed (target) instance is a Linux instance, the sshd config allows for password logon.
- The bastion host, which is chosen by region on the yaml template used to create the bastion host and associated assets, has the SSM and EC2 tools on it and they are active.

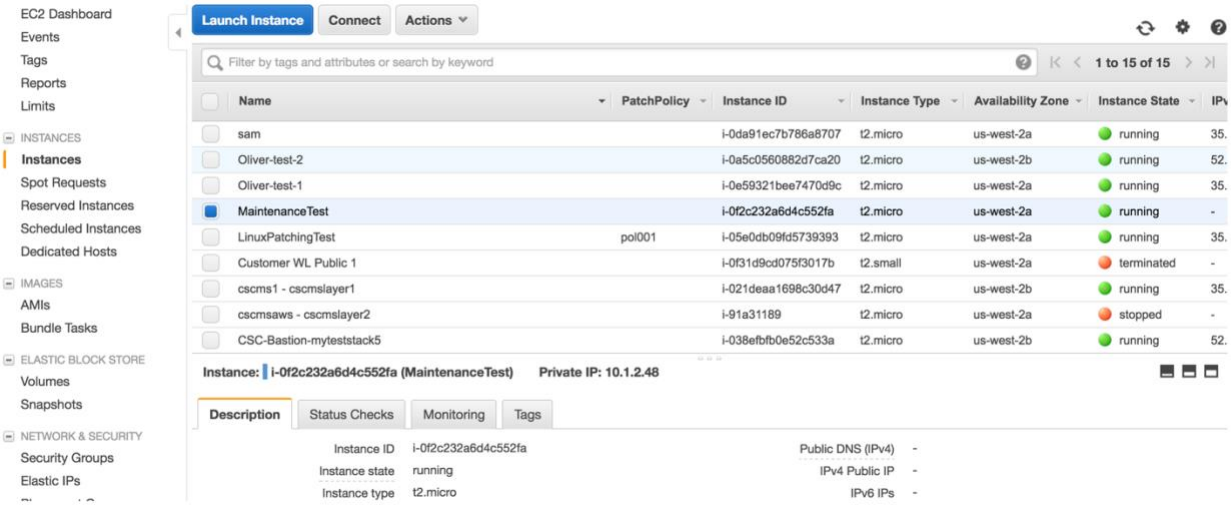
Getting Managed Host Data

The first step is to find the Instance ID of the managed host where you want to connect via the Bastion host. Use the AWS console under EC2 to find the Instance ID as follows:

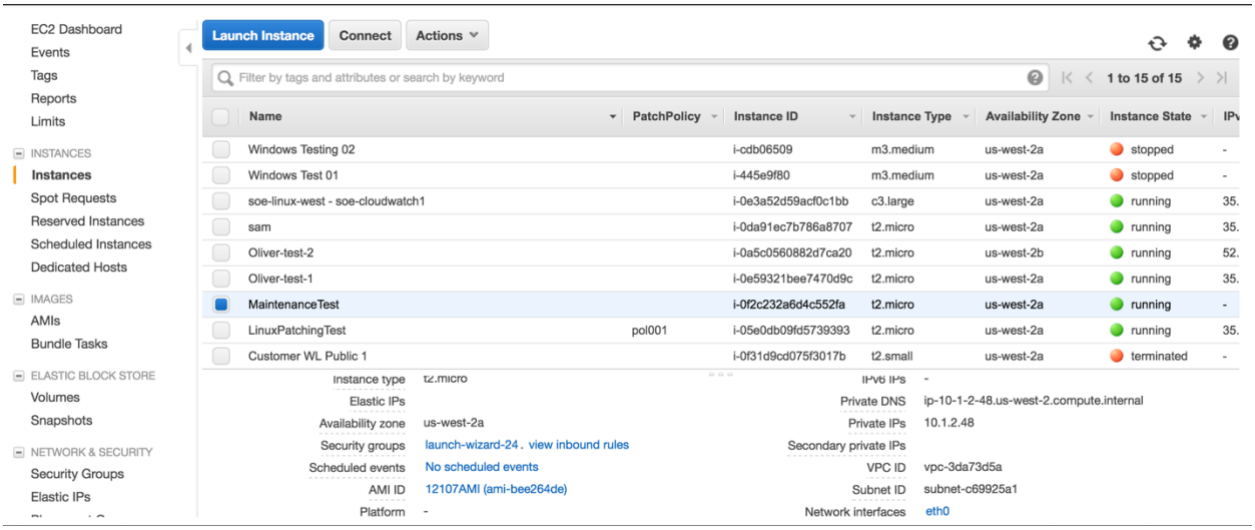
1. From your AWS console, launch **EC2** and then click **Running Instances**.



2. Select your instance and write down the Instance ID.
   In the example, the Instance Name is *MaintenanceTest* and the Instance ID is *i-0f2c232a6d4c552fa*.



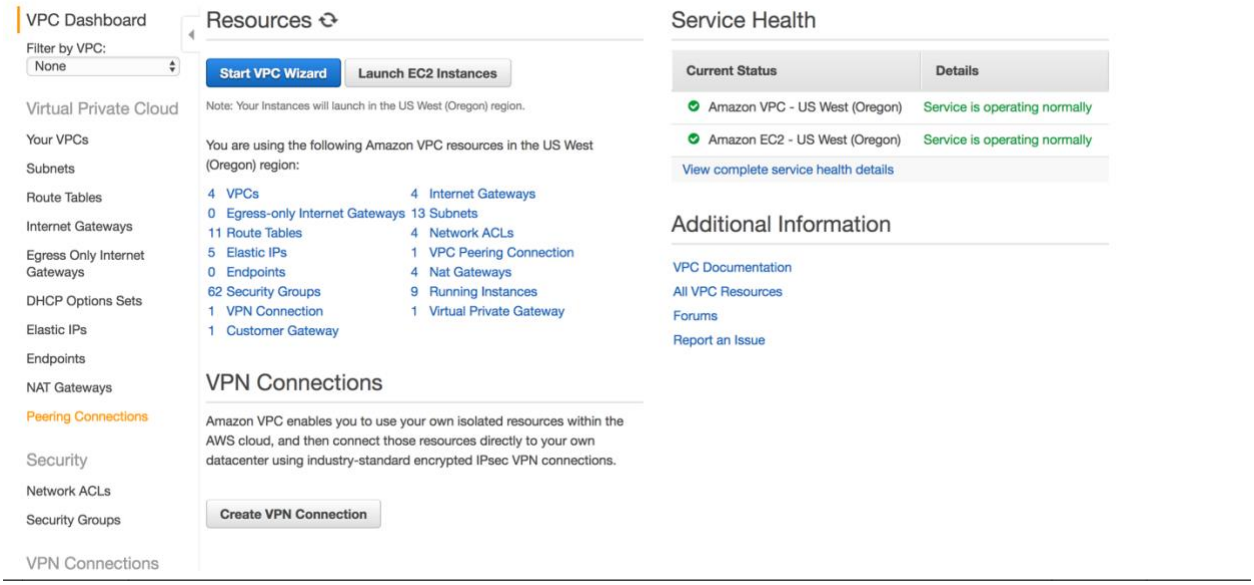
3. Scroll down to view the VPC, Subnet ID, and IP address for this instance.
   In the example, the VPC ID is *vpc-3da73d5a*, the IP address is *10.1.2.48*, and the Subnet ID is *subnet-c69925a1*.



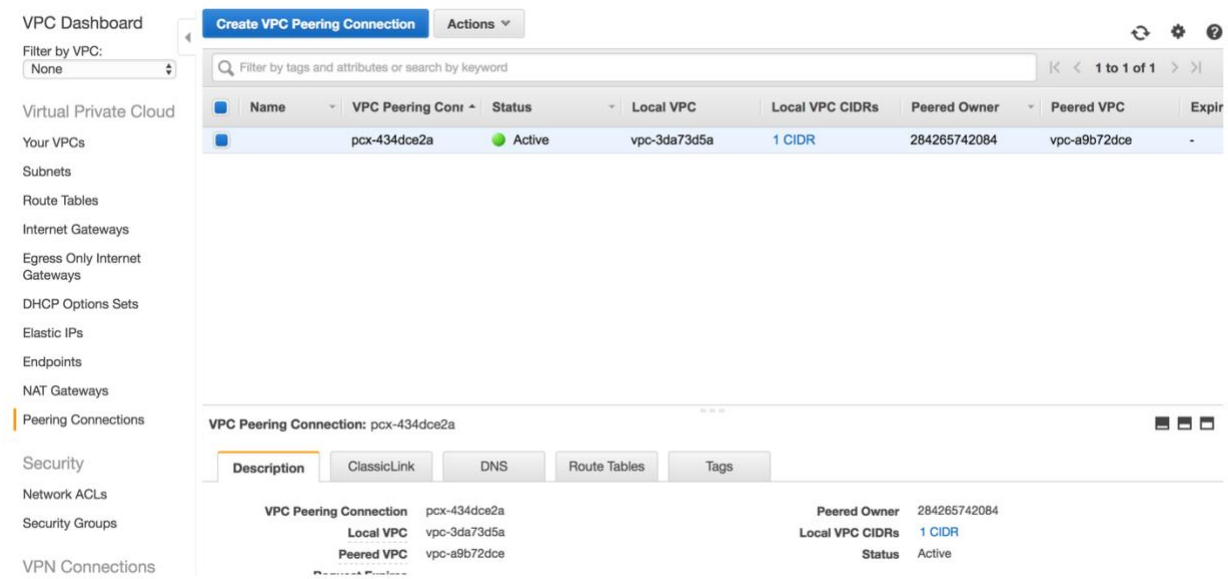
Determining the Subnet ID and VPC for the Bastion Host

The VPC associated with the Bastion host has to be peered with the VPC where the managed instance resides.

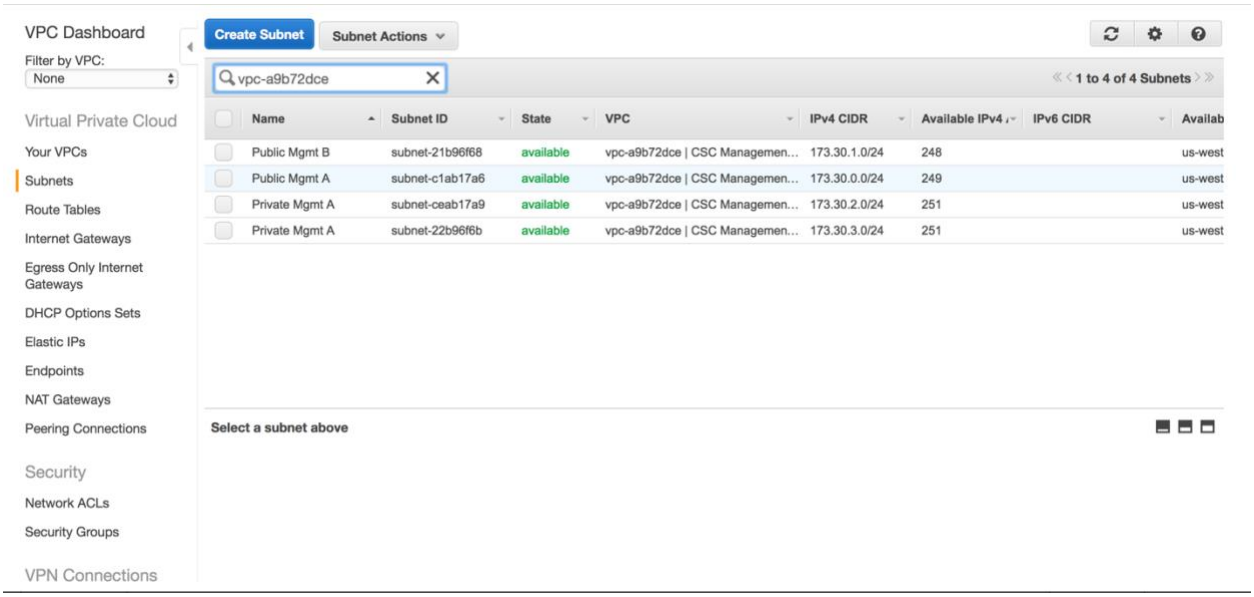
1. In the AWS console, launch the VPC dashboard and click **Peering Connections** on the left.



2. From the list of peering connections, locate the one with the subnet for your managed instance. In the example, there is only one peering connection and this peers the VPC using the subnet to find the managed instance is on *vpc-3da73d5a* with the VPC *vpc-a9b72dce*.



3. Search for subnets associated with the management (peered) VPC *vpc-a9b72dce*.



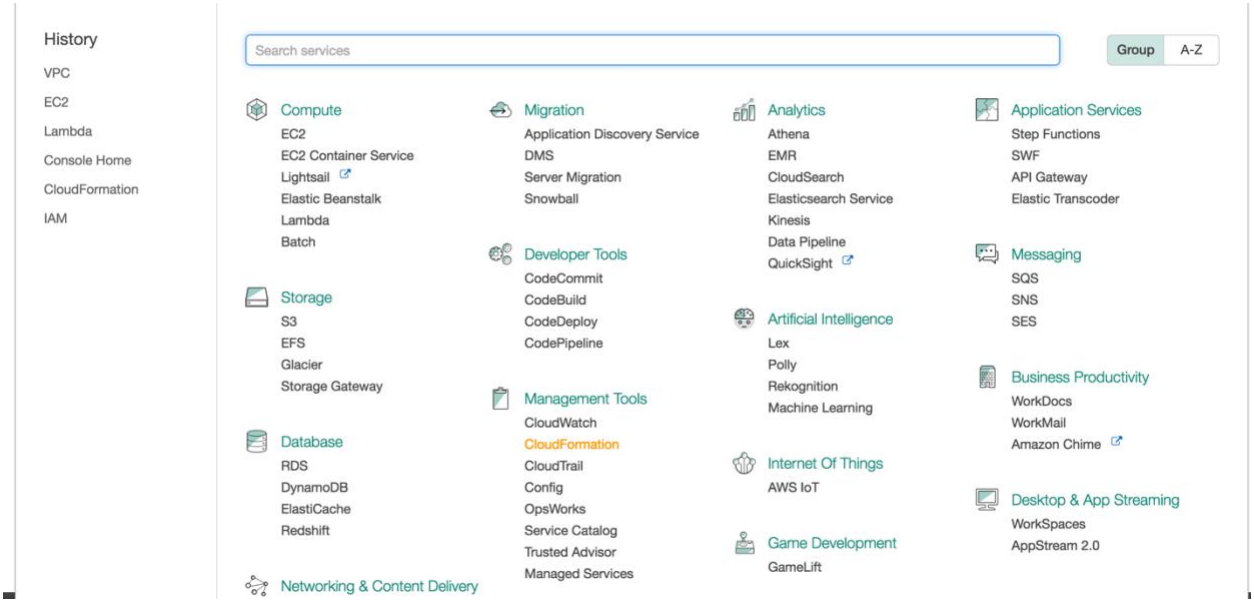
4. Find a subnet that has a route to the managed instance subnet and an available IP address. The example shows selecting Subnet *subnet-21b96f68* that has 248 available IP addresses.

Running the CloudFormation Template

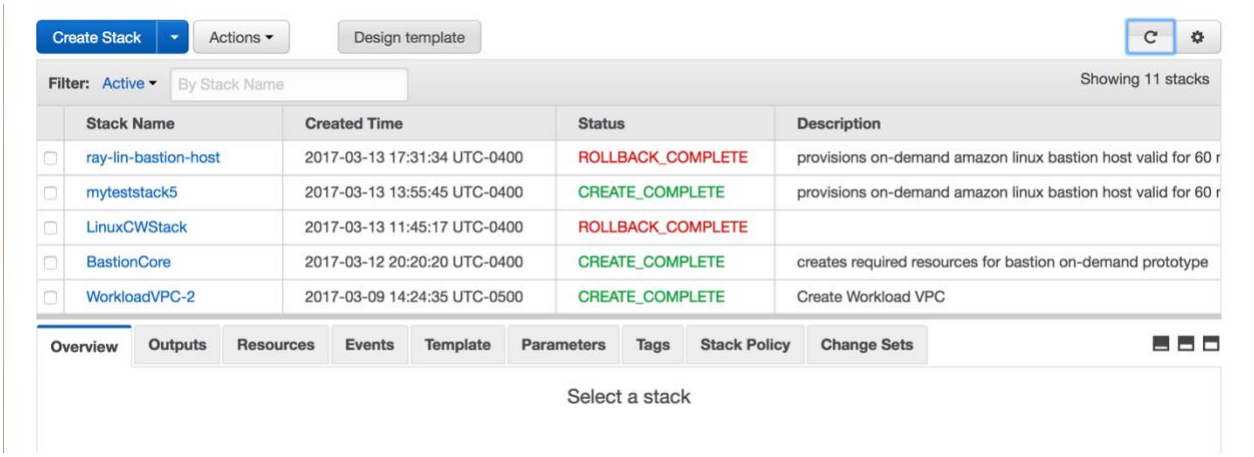
If you are accessing a Windows host, select the CloudFormation template https://github.com/dxchcs/dxcms-cli/blob/master/bastion-service/cloudformation/cft_create_win_bastion.yaml.

If you are accessing a Linux host, select the CloudFormation template https://github.com/dxchcs/dxcms-cli/blob/master/bastion-service/cloudformation/cft_create_linux_bastion.yaml.

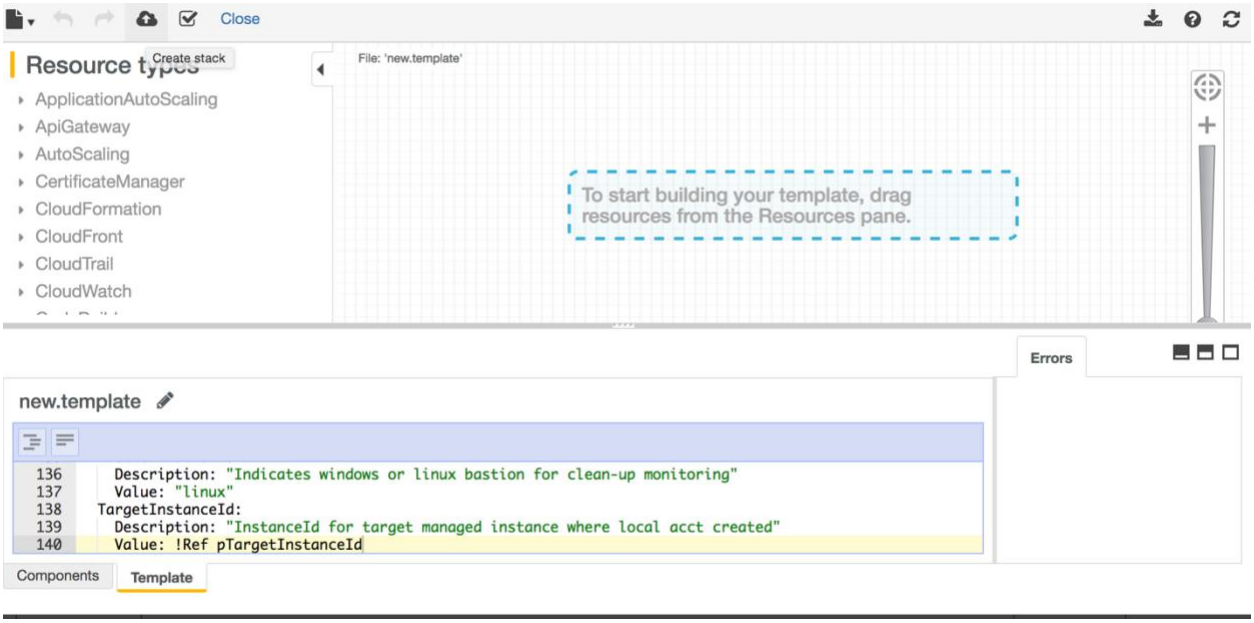
1. Run this template using the CloudFormation tool.
   The example uses the Linux CloudFormation template.



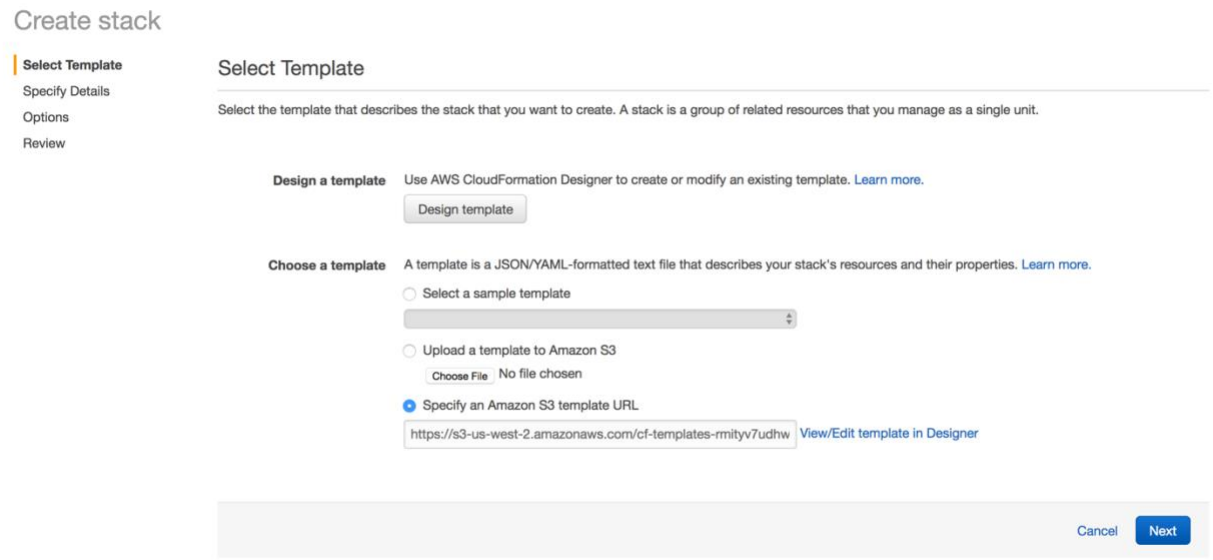
2. Click **Design Template**.



3. In the designer, paste the yaml for Bastion creation (Linux or Windows) into the Template tab.
4. Click the cloud with the up (Create stack) arrow to deploy the template.



5. Click **Next** on the **Stack** page.



Specifying the Stack Details

To specify the stack details:

1. Specify the parameters for the stack as follows:
   Stack name is the name you want to give to the stack. The Bastion instance name is *DXC-Bastion-<StackName>*,

   pClientIp is the IP address of your local machine from which you access the Bastion host,

   pKeyName is the ssh key used to access the Bastion host. You can use the default. In the example,

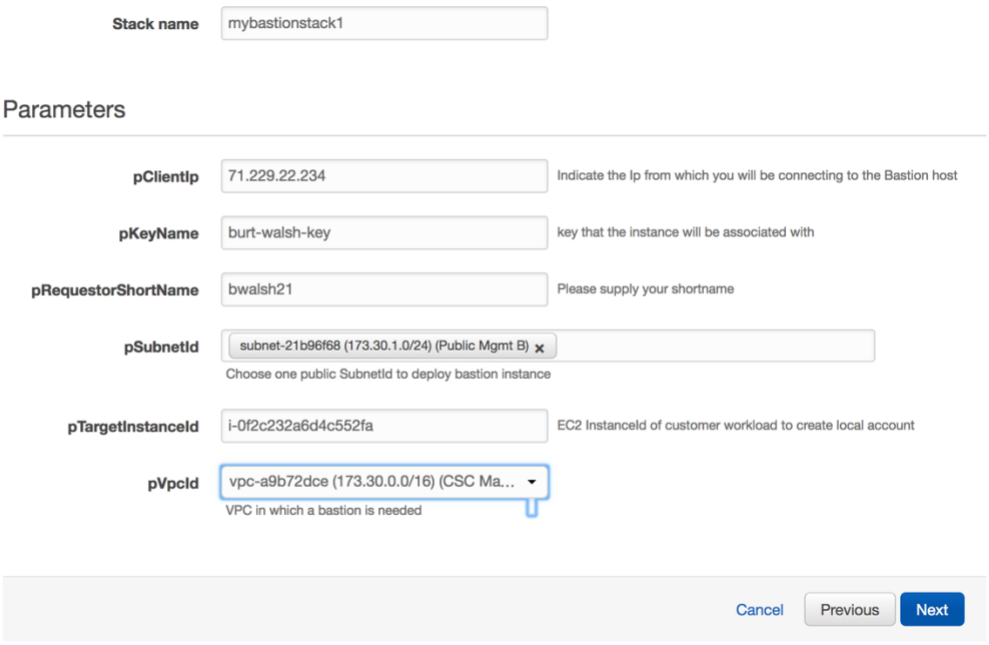
the key name is *burt-walsh-key*,

pRequestorShortName is your email short name,

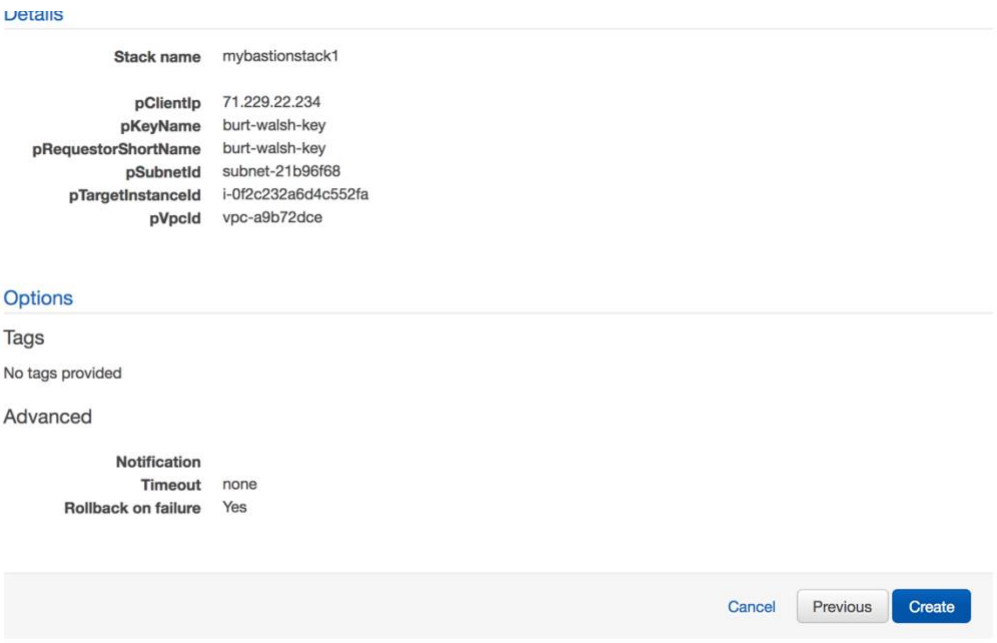
pSubnetId is the subnet for your Bastion host. This ID was determined above and is in a VPC that is paired with the VPC of the managed host you are trying to access.

pTargetInstanceId is the managed instance ID that was determined above.

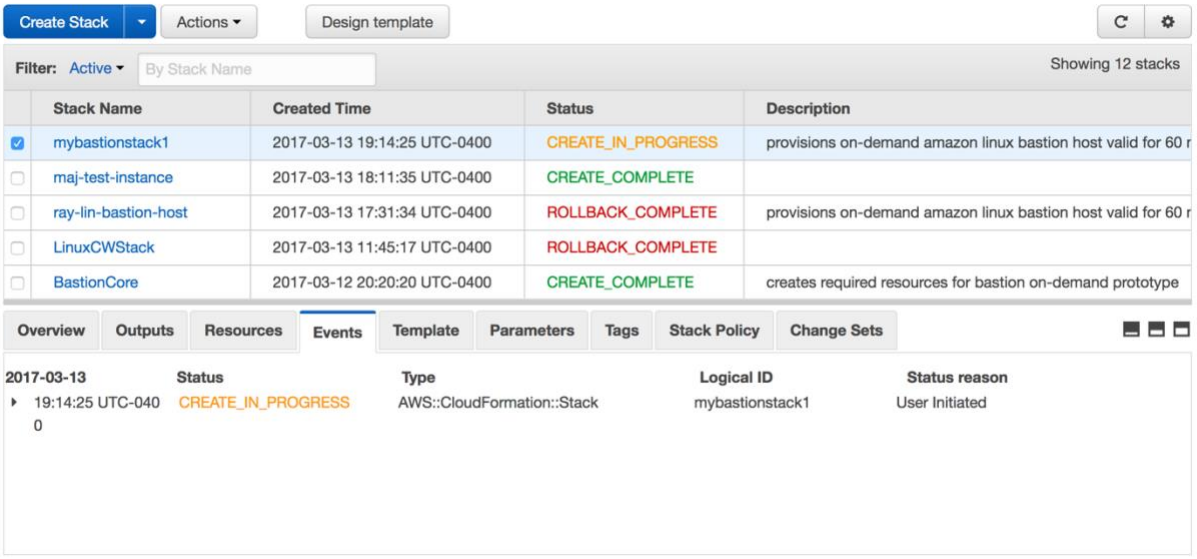
pVpcId is the VPC in which the Bastion instance will be deployed. The pSubnetId (subnet) belongs to this VPC.



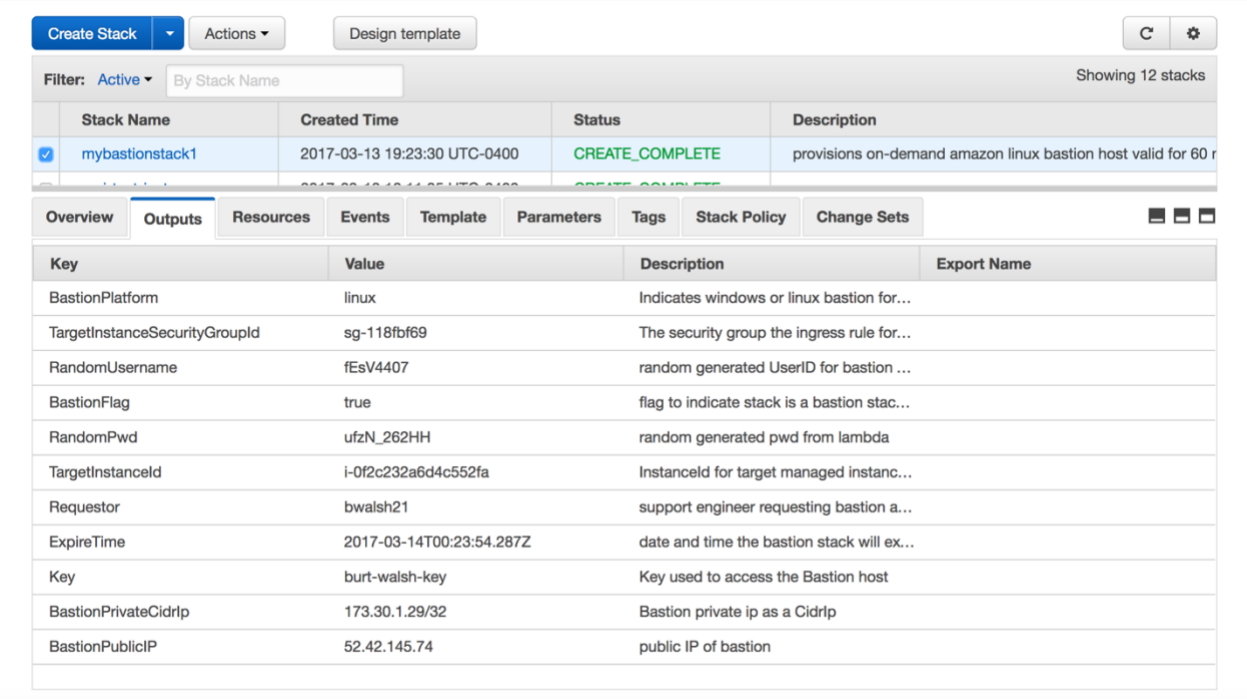
2. After you specify the parameters, click **Next**.
3. Continue from screen to screen and take the defaults until you get to the **Create** screen.
4. On the **Create** screen, click **Create**.



5. Make sure that the creation is in progress.



6. After successful creation, select the stack and click the **Outputs** tab.



Logging into a Linux-Managed Host Through the Bastion Host

This example uses a Linux Bastion host that is paired with a Linux-managed (target) instance.

1. Use PuTTY or a Linux shell to ssh to the Bastion host. This example uses the following output values:
   - RandomUsername fEsV4407
   - RandomPwd ufzN_262HH
   - BastionPublicIp 52.42.145.74
2. If you use ssh, enter the following command:
   `ssh fEsV4407@52.42.145.74`
3. Enter the password (*ufzN_262HH*) at the prompt.
4. After you have connected to the Bastion host, ssh to the managed host with the following command:
   `ssh fEsV4407@10.1.2.48`
5. Enter the password (*ufzN_262HH*) at the prompt.

Note: You determined the IP address of the managed instance when you looked at the running instance above.

The following image shows the steps on the Linux shell:

```
Last login: Mon Mar 13 18:19:04 on ttys000
asalazar:~ bwalsh$ ssh fEsV4407@52.42.145.74
The authenticity of host '52.42.145.74 (52.42.145.74)' can't be established.
RSA key fingerprint is 57:66:78:47:70:94:f6:93:37:18:dd:24:84:73:14:d3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.42.145.74' (RSA) to the list of known hosts.
fEsV4407@52.42.145.74's password:

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

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
[fEsV4407@ip-173-30-1-29 ~]$ ssh fEsV4407@10.1.2.48
The authenticity of host '10.1.2.48 (10.1.2.48)' can't be established.
ECDSA key fingerprint is b3:23:ad:7e:0a:3f:8f:d0:79:e5:8e:6f:4f:56:2a:98.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.2.48' (ECDSA) to the list of known hosts.
fEsV4407@10.1.2.48's password:

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

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
6 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[fEsV4407@ip-10-1-2-48 ~]$
```

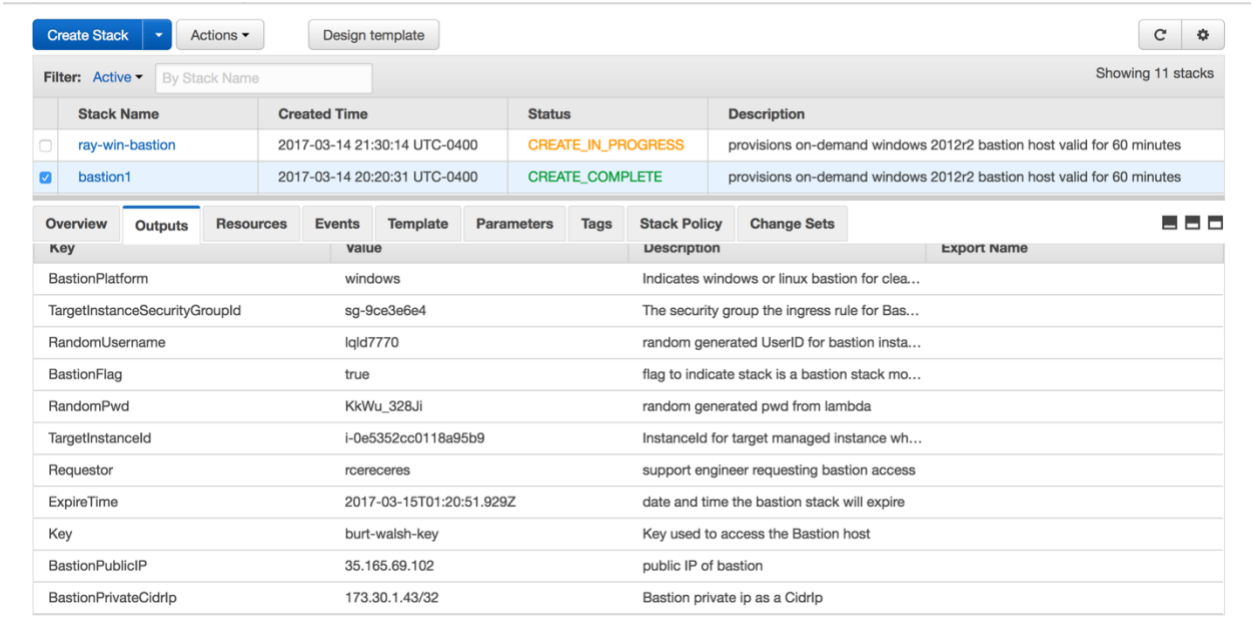
Logging into a Windows-Managed Host Through the Bastion Host

This example uses a Windows Bastion host that is paired with a Windows-managed (target) Instance. Use a Remote Desktop Client (RDP). For Windows, RDP to the Bastion host and then to the managed host.

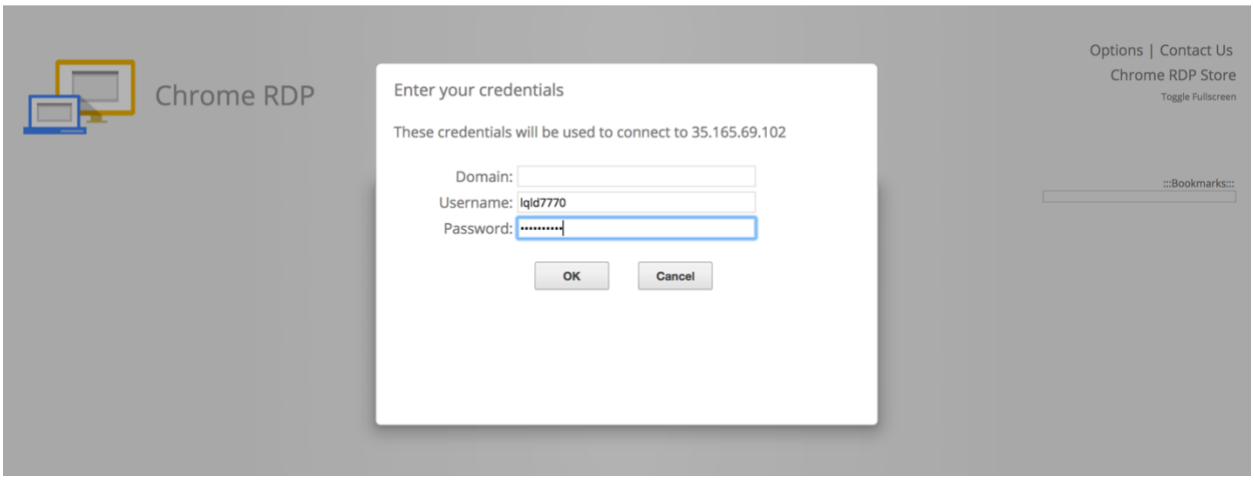
This example uses the following output values:

- RandomUsername lqld7770
- RandomPwd KkWu_328Ji
- BastionPublicIp 35.165.69.102

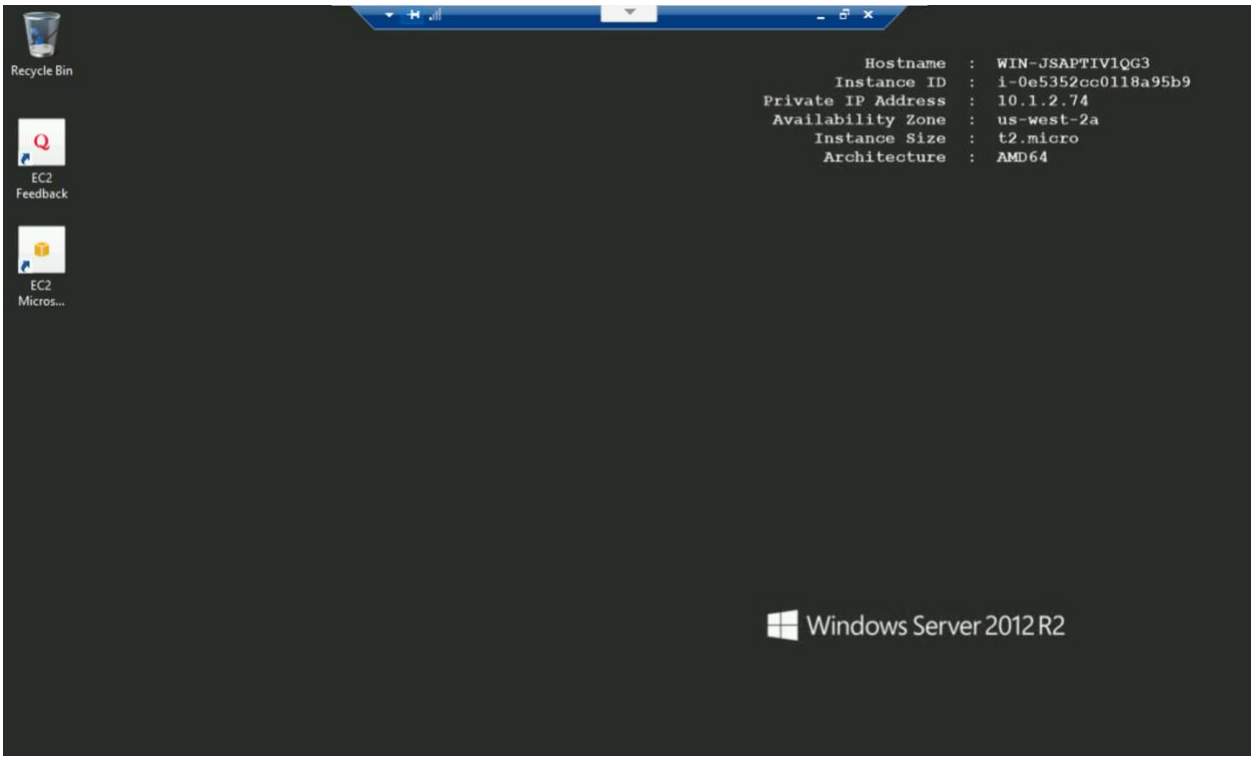
The following stack description shows these values:



1. From your machine, enter the username and password into the RDP client.



2. On the Bastion host, use the RDP client on the host to access the managed (target) host with the same username and password.



Provisioning Workloads

15

This section describes how instances are provisioned using the provided CloudFormation templates.

Prerequisites

AMIs need to exist in the customer account for each region the customer plans to use. The following AMIs are currently supported by the Offering Build Team: RHEL 6.7, RHEL 7.2, Windows 2012 R2 and Windows 2016. See the AMI chapter for instructions on how to obtain the images.

If you have not run the Master template or successfully configured Gold Managed Services, you (the Delivery Engineer) must go back and complete those tasks or the instances will not be properly managed.

Provisioning Linux Instances

Permissions

This process is performed by a Delivery Engineer from the AWS Console. The *Delivery Power User* role created automatically by the Master template should be used to execute the templates.

Overview

Creating a new Linux instance includes the following steps:

1. Selecting a location for the instance: VPC, Subnet, and a Public IP option.
2. Selecting an OS Type or a Custom AMI along with Security Groups and a key pair.
3. Specifying administrative tag information.
4. Specifying services overrides information.
5. Specifying root volume parameters.
6. Specifying optional volume parameters.

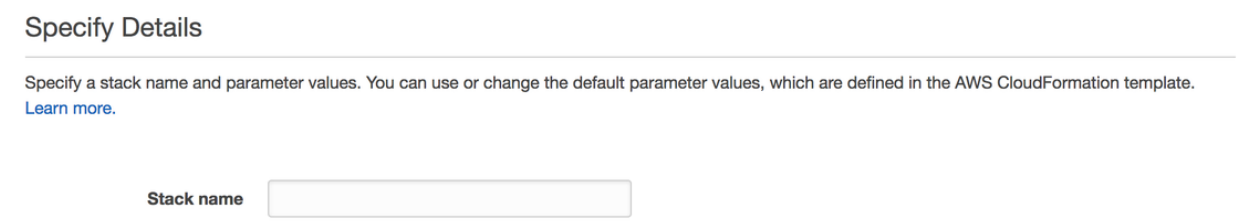
Creating the Stack

To create the stack:

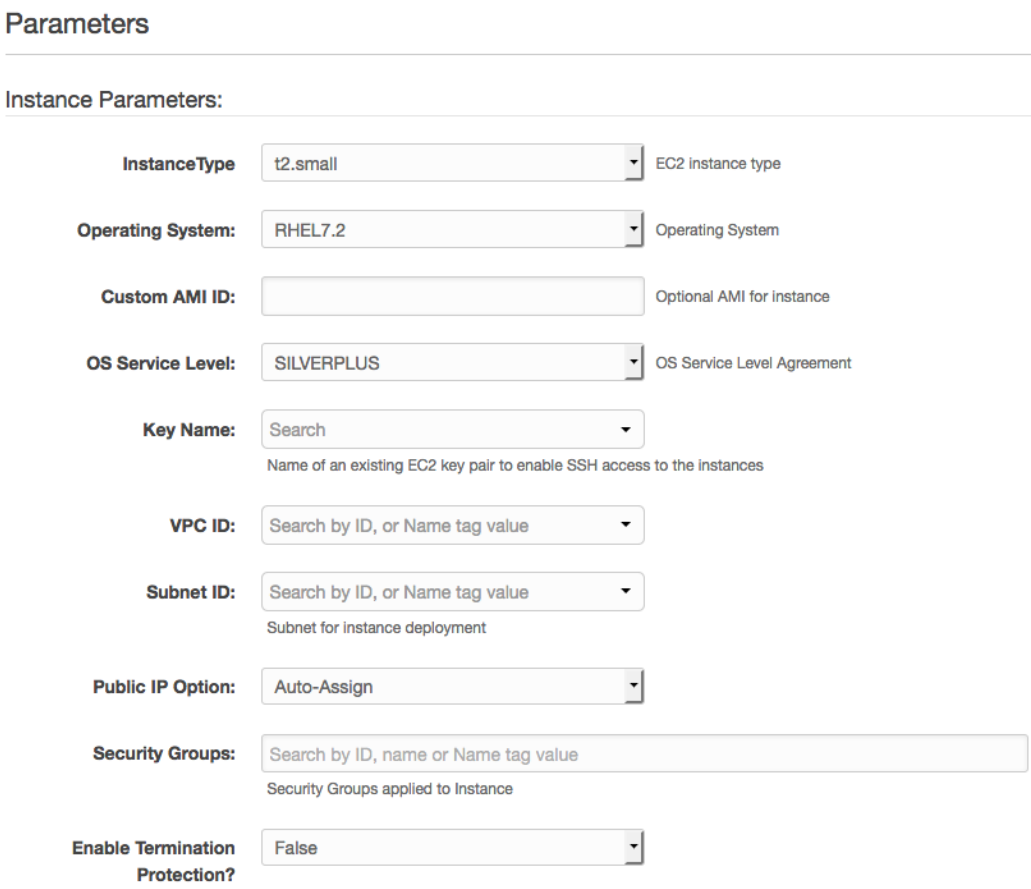
1. Log into the customer's AWS account in the AWS console.
2. Under **Services**, select **CloudFormation**.
3. Click **Create Stack**.
4. Select **Specify an Amazon S3 template URL**.
5. Enter the path in S3 to the CloudFormation template named **simple-workload-amazon-linux.json or simple-workload-rhel-linux.json**.
   For example: *https://s3.amazonaws.com/dxc.customer.config-211682634048-us-west-2/cloudformation/simple-workload-rhel-linux.json*
6. Click **Next**
7. Follow the instructions outlined below.

The template contains several sections. Although they are continuous on the page, they are broken apart here.

1. Enter a unique **Stack name.**



2. Specifying the main instance parameters:

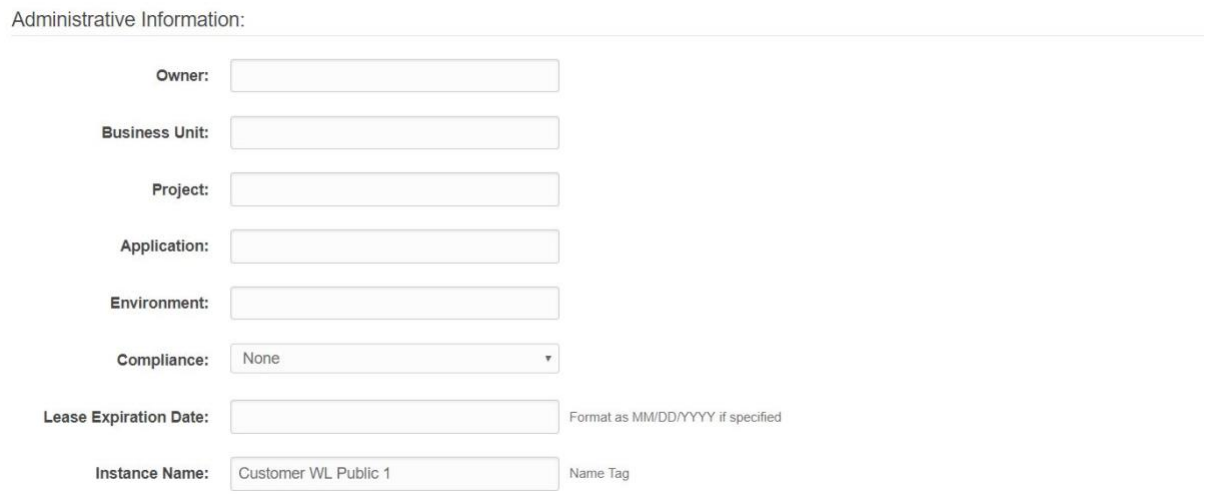


The following parameters control the main configuration of the instance:

Parameter	Description
Instance Type	Select one of the available Instance Types. This parameter controls the amount of memory and number of CPUs allocated to the instance.

Operating System	Select one of the designated operating systems (currently RHEL6.7 or RHEL7.2) or select the **Custom.AMI** option to specify an AMI of your choice. If you select one of the designated operating systems, the AMI latest SOE for that OS will be used.
Custom AMI ID	If the **Custom.AMI** option was selected for the Operating System, then specify the desired AMI here.
OS Service Level	Select either *SILVERPLUS* or *GOLD* based upon your service subscription. (*Note*: *If the OS Service Level selected is different than what is associated to the account, issues will result with the stack deployment*)
Key Name	Enter the name of the SSH key to install on the new instance.
VPC ID	Specify the ID of the VPC where the Instance will be launched, which is typically one of the Customer Workload VPCs created under the customer's account.
Subnet ID	Specify the Subnet ID of the subnet where the Instance will be launched.
Public IP Option	Select one of these options: Auto-Assign - A public IP will be assigned automatically by AWS. Elastic-IP - An Elastic IP will be allocated and assigned to the Instance. None - No public IP will be assigned. If you select *None* and the Instance resides in a *public* subnet (one that has an Internet Gateway connected), then CloudWatch metrics and logs will not be available because there is no connection to the public Internet. Instances in a *private* subnet (no Internet gateway) are able to connect to CloudWatch through the NAT gateway and thus do not necessarily need a public IP.
Security Groups	Specify the names of the Security Groups that will be assigned to the Instance. Select the Security Group carefully. A wide-open range of 0.0.0.0/0 allows anyone with access to an Instance that has a Public IP. Allowable address ranges should be limited to Corporate networks or allow access only from designated networks. Ingress port 443 (HTTPS) must be opened on the Instance for the Crowdstrike Falconhost agent to communicate back to the Crowdstrike portal.
Enable Termination Protection	Select *True* if you want to prevent accidental termination of the Instance from the AWS Console.

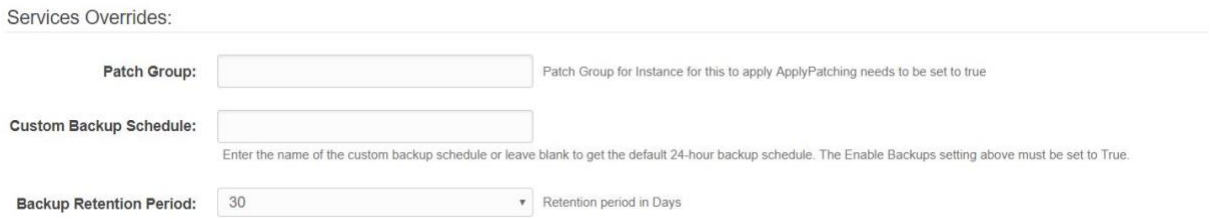
3. Define the Administrative Tag Parameters



The business tags used by CloudCheckr and ServiceNow are:

Parameter	Description
Project	Specify which project the instance supports.
Owner	Type the name of the owner of the instance, this is typically the requestor.
Lease Expiration Date	Specify the date when the VM should be stopped and released.
Instance Name	A friendly name for the instance, which will be copied to the instance's volumes and snapshots.
Environment	Select an environment the instance is bound to, for example: Dev, Prod, QA, or Staging.
Compliance	Select a compliance tag, for example: HIPPA or ITAR.
Application	Specify which application the instance supports.

4. Define parameters for Service Overrides:



Provide the following Service Override Parameters **only if you would like to modify the Patch Group, Custom Backup Schedule and Backup Retention Period values** that were defined in DXC Main Template. This will override the values that were defined in the DXC Main template.

Parameter	Description
Backup Retention Period	Specify the number of days that an individual backup is maintained. Possible values are 30, 60, and 90 days.
Custom Backup Schedule	Enter the name of a custom backup schedule if the volumes for the instance should be backed up using a custom schedule. Leave this field blank if you want to use the default backup schedule (once every 24 hours).
Patch Group	Specify a Patch Group for the instance.

5. Define the Root (boot) volume disk parameters:

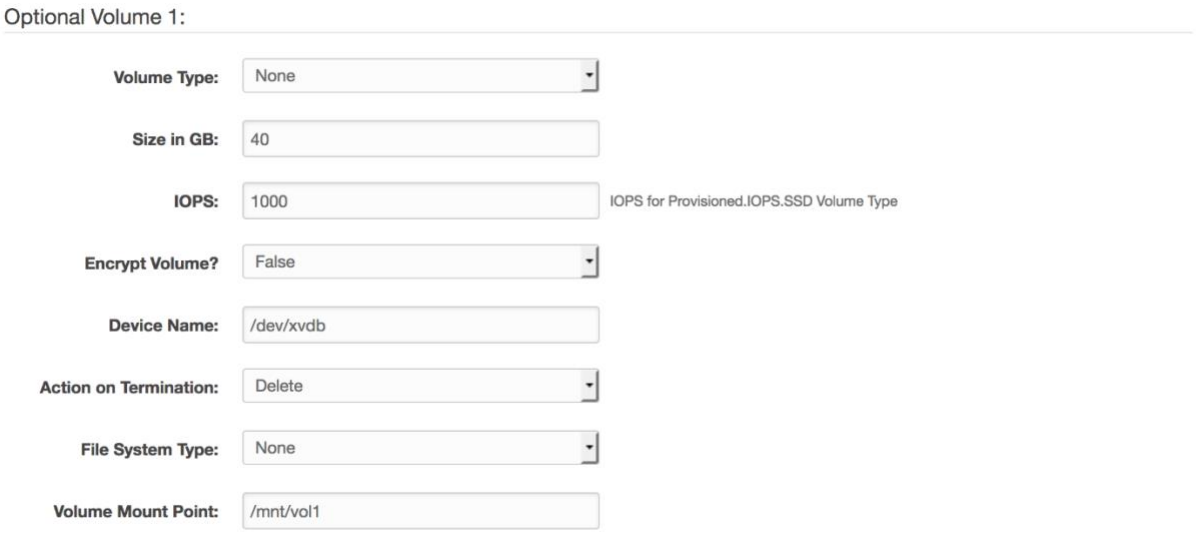
The parameters for the root volume are:

Parameter	Description
Root Volume Type	The options are: • Magnetic • Provisioned.IOPS.SSD • General.Purpose.SSD • Throughput.Optimized.HDD • Cold.HDD
Root Volume Size in GB	The root volume size, in gigabytes.
Root Volume IOPS	Used only for Provisioned.IOPS.SSD. This parameter represents the number of I/O operations per second that the volume can support. The range is from *100* to *2000*.
Root Volume Delete on Termination	If set to *True*, the root volume is deleted when the instance terminates; otherwise, the volume remains. There is no automatic cleanup of volume storage. If you select *False*, the user is responsible for eventual deletion of the volume.

Root Volume Encryption	If set to *True*, an AMI with a tag "encrypted" with value "true" will be used, for AMIs of the Operating System parameter selected. If *False*, an AMI with a tag "encrypted" with value "true" will be used. If you are using a Custom AMI (Operating System = Custom.AMI, and the Custom AMI ID parameter is not blank), then this parameter is ignored, and the root volume will only be encrypted if the AMI you specify has an encrypted root volume.

6. Specify additional volumes:

The additional two volumes have identical parameters.



The parameters for the two optional volumes are:

Parameter	Description
Volume Type	The options are: • **None** - No additional volume is created • **Magnetic** • **Provisioned.IOPS.SSD** • **General.Purpose.SSD** • **Throughput.Optimized.HDD** • **Cold.HDD**
Size in GB	Size in gigabytes of each volume.
IOPS	Used only for Provisioned.IOPS.SSD. This parameter represents the number of I/O operations per second that the volume can support. The range is from *100* to *2000*.
Encrypt Volume	If you select *True*, the "default" encryption key is used. There is currently no support for a custom encryption key.
Device Name	Enter the device name where the volume will be attached. The default "next available" device names are shown as defaults.

Action On Termination	There is no automatic cleanup of volume storage. If you select *Retain* or *Snapshot*, the user is responsible for eventual deletion of the volume or snapshots. • **Delete** - Delete the volume when the instance is released. • **Retain** - Retain the volume when the instance is released. • **Snapshot** - Maintain a snapshot of the volume when the instance is released.
File System Type	Set this to one of the possible EXT file system types or set to *None* and no file system will be formatted. If you want a file system other than EXT2, EXT3, or EXT4, select *None*. You will then be required to manually format and mount the volume.
Volume Mount Point	Specify the mount point of the newly created file system if one of the EXT options is selected.

When the instance is created, the following tags are created:

Tag	Description
Name	Per user input
Owner	Per user input
Department	Per user input
Project	Per user input
Application	Per user input
Environment	Per user input
Compliance	Per user input
LeaseExpirationDate	Per user input
PatchPolicy	Per user input
PatchGroup	Per user input
BackupSchedule	Per user input
OSName	Per user input
RetentionPeriod	Per user input (*30*/*60*/*90*)
InstanceName	Per user input (same as Name tag)

Backup	Per user input (*True*/*False*)
ApplyEndPointProtection	Per user input (*True*/*False*)
ApplyLogging	Per user input (*True*/*False*)
ApplyPatching	Per user input (*True*/*False*)
ApplyMonitoring	Per user input (*True*/*False*)
DXCProduct	QuickSilver

For volumes created as part of the instance, the following tags are created automatically:

Tag	Description
Name	Constructed from the *InstanceName* tag and the device attach point (for example, *MyInstance-/dev/xbdb*).
InstanceName	From instance tags
Owner	From instance tags
Department	From instance tags
Project	From instance tags
Application	From instance tags
Environment	From instance tags
MountPoint	Per user input (if a File System is specified)

For EBS Snapshots created as part of the instance, the following tags are created automatically:

Tag	Description
Name	From instance tags
Owner	From instance tags
Department	From instance tags
Project	From instance tags
Environment	From instance tags

DeleteOn	From instance tags

Provisioning Windows Instances

Permissions

This process is performed by a Delivery Engineer from the AWS Console. The *Delivery Power User* role created automatically by the Master template should be used to execute the templates.

Overview

Creating a new Windows instance includes the following:

- Selecting a location for the instance: VPC, Subnet, and a Public IP option.
- Selecting an OS Type or a Custom AMI along with Security Groups and a key pair.
- Specifying administrative tag information.
- Specifying root volume parameters.
- Specifying optional volume parameters.

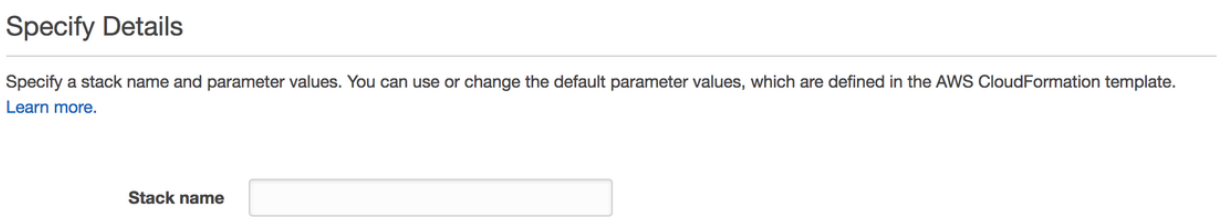
Creating the Stack

To create the stack:

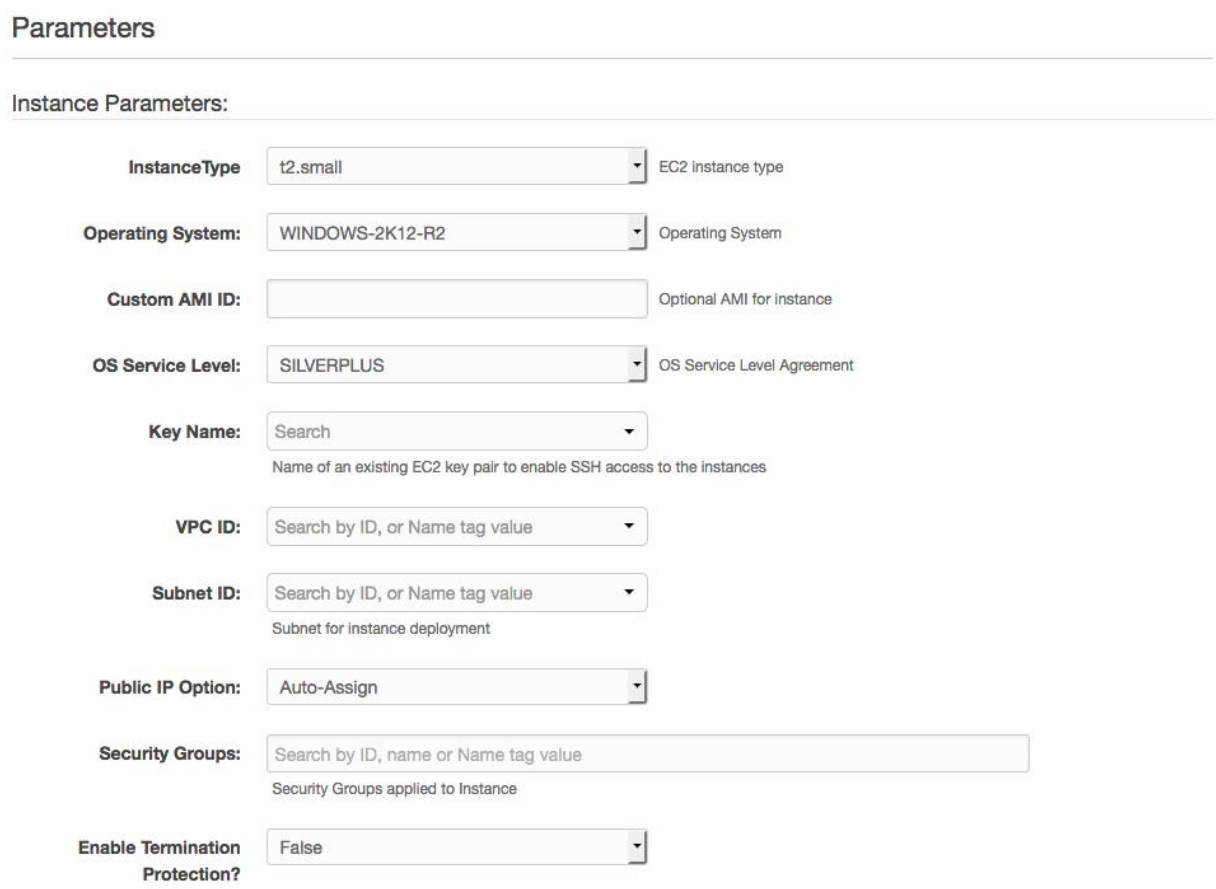
1. Log into the customer's AWS account in the AWS console.
2. Under **Services**, select **CloudFormation**.
3. Click **Create Stack**.
4. Select **Specify an Amazon S3 template URL**.
5. Enter the path in S3 to the CloudFormation template named **simple-workload-windows.json**.
   For example: https://s3.amazonaws.com/cschcs-ms-deployment/cloudformation/simple-workload-windows.json
6. Click **Next**.
7. Follow the instructions outlined below.

The template contains several sections. Although they are continuous on the page, they are broken apart here.

1. Enter a unique **Stack name**.



2. Specify the main instance parameters.

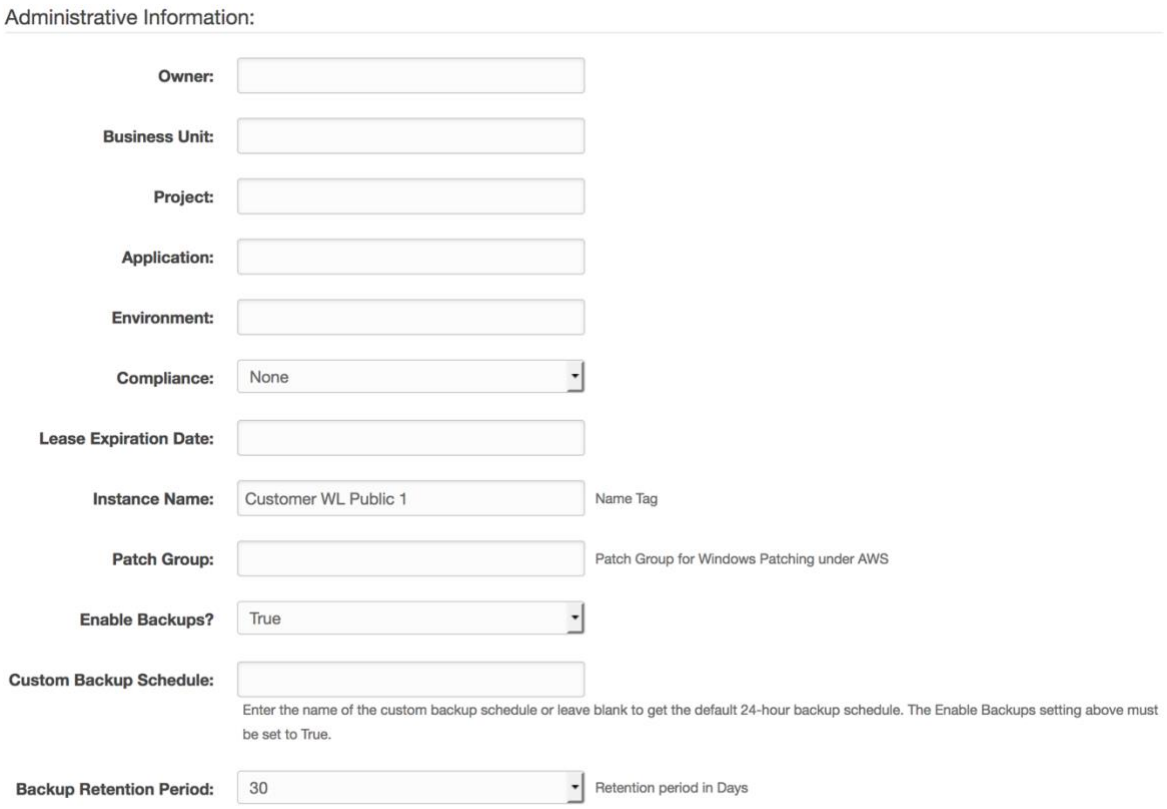


These parameters control the main configuration of the instance:

Parameter	Description
Instance Type	Select one of the available Instance Types. This parameter controls the amount of memory and number of CPUs allocated to the instance.
Operating System	Select one of the designated Windows operating systems or select the Custom.AMI option to specify an AMI of your choice. If you select one of the designated operating systems, the AMI latest SOE for that OS will be used.
Custom AMI ID	If the **Custom.AMI** option was selected for the Operating System, then specify the desired AMI here.
OS Service Level	Select either *SILVERPLUS* or *GOLD* based upon your service subscription. (*Note*: *If the OS Service Level selected is different than what is associated to the account, issues will result with the stack deployment*)
Key Name	Enter the name of the SSH key to install on the new instance.
VPC ID	Specify the ID of the VPC where the Instance will be launched, which is typically one of the Customer Workload VPCs created under the customer's account.

Subnet ID	Specify the Subnet ID of the subnet where the instance will be launched.
Public IP Option	Select one of these options: Auto-Assign - A public IP will be assigned automatically by AWS. Elastic-IP - An Elastic IP will be allocated and assigned to the Instance. None - No public IP will be assigned. If you select *None* and the Instance resides in a *public* subnet (one that has an Internet Gateway connected), then CloudWatch metrics and logs will not be available because there is no connection to the public Internet. Instances in a *private* subnet (no Internet gateway) are able to connect to CloudWatch through the NAT gateway and thus do not necessarily need a public IP.
Security Groups	Specify the names of the Security Groups that will be assigned to the instance. Select the Security Group carefully. A wide-open range of 0.0.0.0/0 allows anyone with access to an instance that has a Public IP. Allowable address ranges should be limited to corporate networks or allow access only from designated networks.
Enable Termination Protection	Select *True* if you want to prevent accidental termination of the Instance from the AWS Console.
CrowdStrike CID	The key for the CrowdStrike software that will run on the instance.

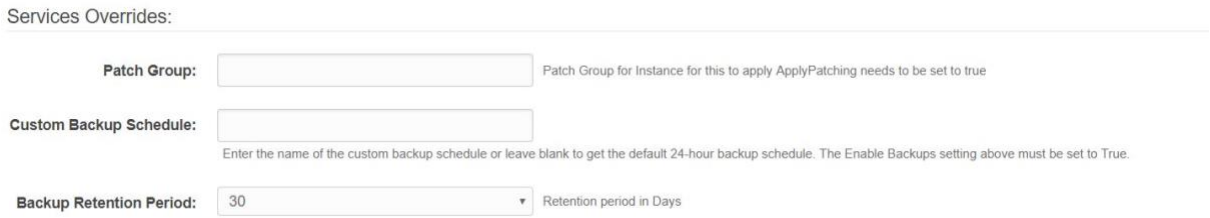
3. Define the Administrative Tag parameters.



For EBS Snapshots created as part of the instance, the following tags are created automatically:

Tag	Description
Name	From instance tags
Owner	From instance tags
Department	From instance tags
Project	From instance tags
Environment	From instance tags
DeleteOn	From instance tags

4. Define the Services Overrides parameters.



This will override the values that were defined in the DXC Main template.

Parameter	Description
Patch Group	Select a Patch Group for the instance.
Custom Backup Schedule	Enter the name of a custom backup schedule if the volumes for the Instance should be backed up using a custom schedule. Leave this blank to get the default backup schedule (once every 24 hours).
Backup Retention Period	Number of days that an individual backup is maintained. Possible values are 30, 60, and 90 days.

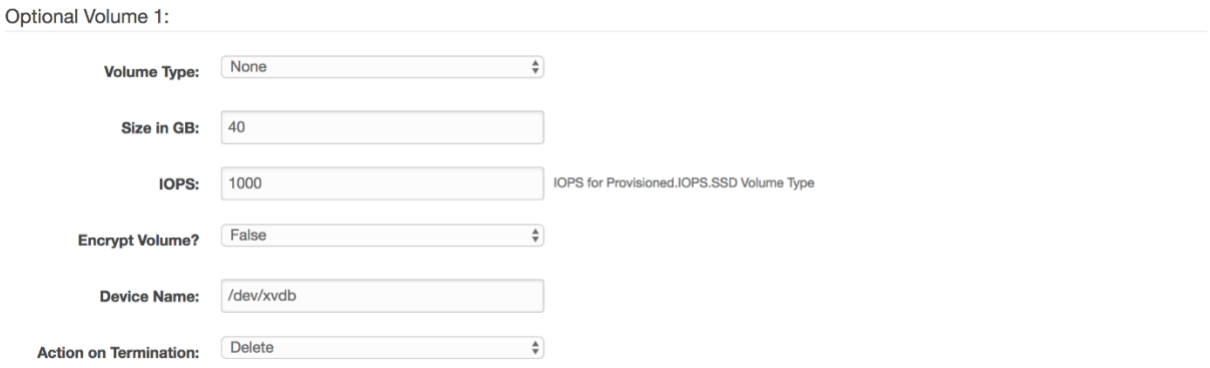
Note: It is mandatory to specify a Patch Group if Custom AMI (RHEL, AMAZON, Windows) is selected under the operating system option.

5. Define the Root (boot) Volume Disk parameters.

The parameters for the root volume are:

Parameter	Description
Volume Type	The options are: • **None** - No additional volume is created • **Magnetic** • **Provisioned.IOPS.SSD** • **General.Purpose.SSD** • **Throughput.Optimized.HDD** • **Cold.HDD**
Size in GB	The root volume size, in gigabytes.
IOPS	Used only for Provisioned.IOPS.SSD. This parameter represents the number of I/O operations per second that the volume can support. The range is from *100* to *2000*.
Delete On Termination	If set to *True*, the root volume is deleted when the instance terminate; otherwise, the volume remains. There is no automatic cleanup of volume storage. If you select *False*, the user is responsible for eventual deletion of the volume.

6. Specify additional volumes



The additional two volumes have identical parameters.

The parameters for the two optional volumes are:

Parameter	Description
Volume Type	The options are: • **None** - No additional volume is created • **Magnetic** • **Provisioned.IOPS.SSD** • **General.Purpose.SSD** • **Throughput.Optimized.HDD** • **Cold.HDD**
Size in GB	Size in gigabytes of each volume.
IOPS	Used only for Provisioned.IOPS.SSD. This parameter represents the number of I/O operations per second that the volume can support. The range is from *100* to *2000*.
Encrypt Volume	If you select *True*, the "default" encryption key is used. There is currently no support for a custom encryption key.
Device Name	Enter the device name where the volume will be attached. The default "next available" device names are shown as defaults.
Action On Termination	There is no automatic cleanup of volume storage. If you select *Retain* or *Snapshot*, the user is responsible for eventual deletion of the volume or snapshots. • **Delete** - Delete the volume when the instance is released. • **Retain** - Retain the volume when the instance is released. • **Snapshot** - Maintain a snapshot of the volume when the instance is released.

Note: The creation process updates the hostname of the Windows instance to the first 15 characters of the Instance ID. Currently instances are not joined to a domain and the Restart-Computer does not accommodate a domain.

AWS Config Rules

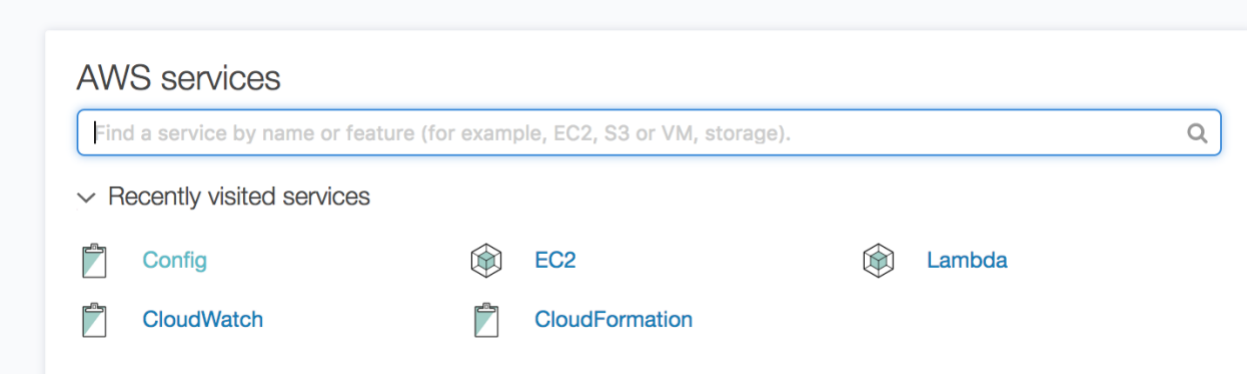
AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. This includes how the resources are related to one another and how they were configured in the past so that you can see how the configurations and relationships change over time.

An AWS *resource* is an entity you can work with in AWS, such as an Amazon Elastic Compute Cloud (EC2) instance, an Amazon Elastic Block Store (EBS) volume, a security group, or an Amazon Virtual Private Cloud (VPC), for example.

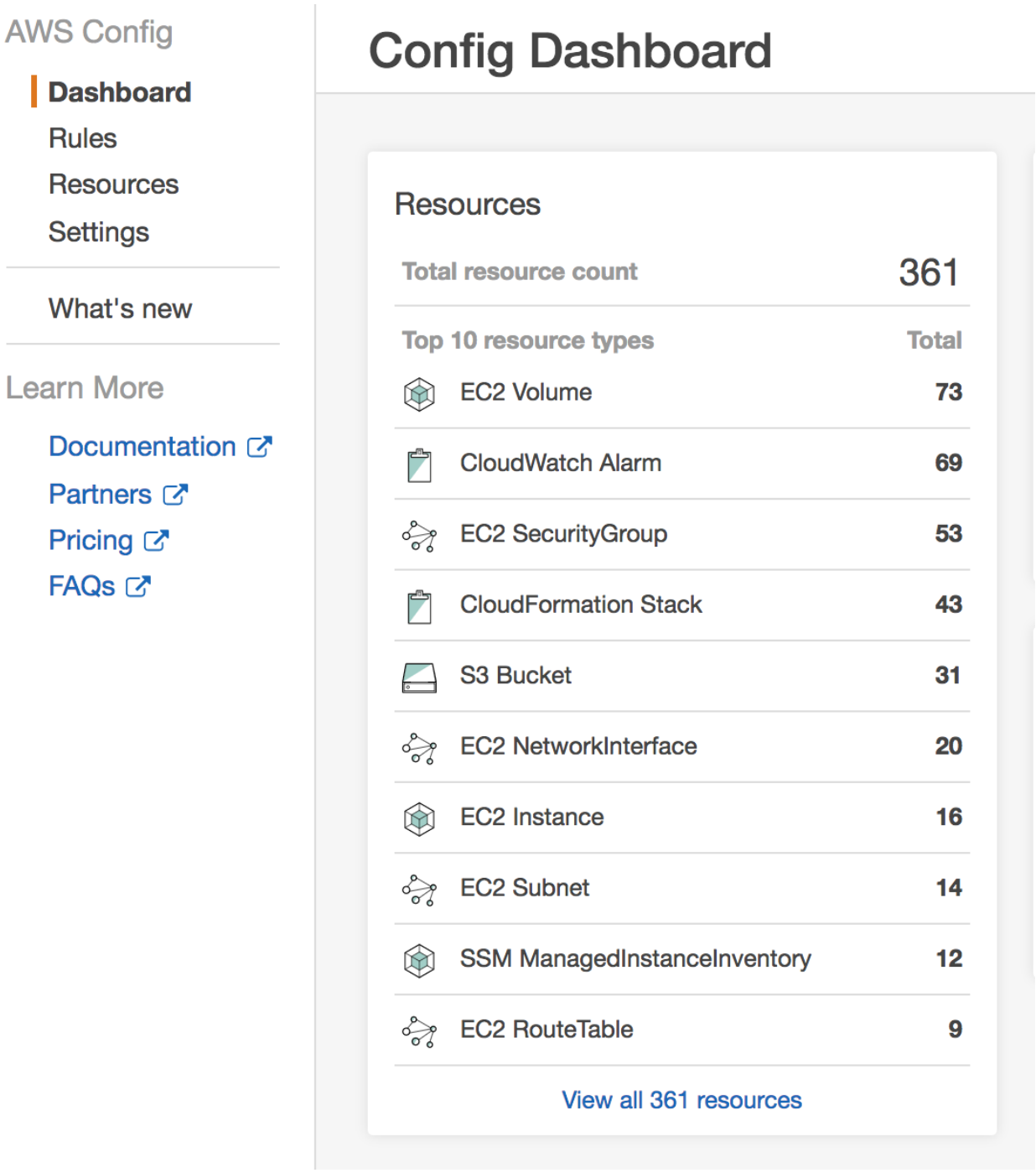
AWS Config Rule for Backup enabled instances

Complete the following steps to ensure all instances have backup enabled and either have a custom backup schedule or the default backup schedule.

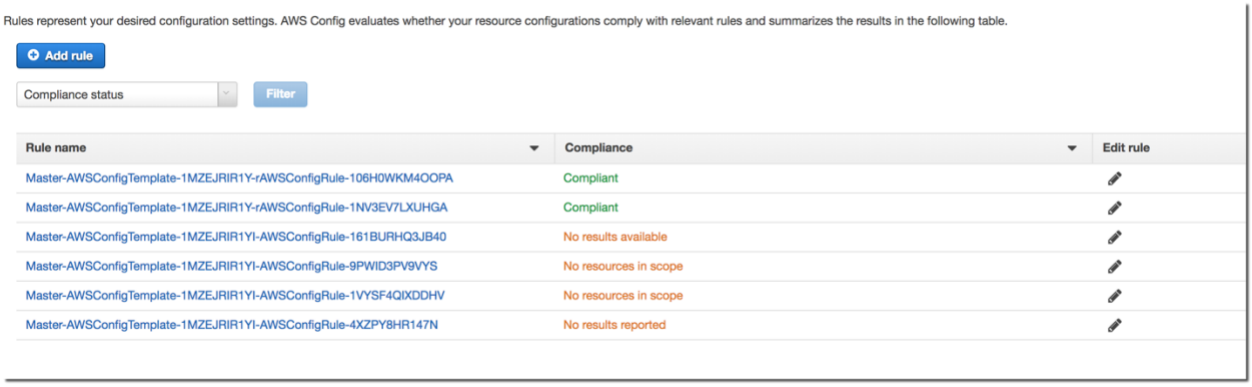
1. Go to AWS Services and click **Config**.



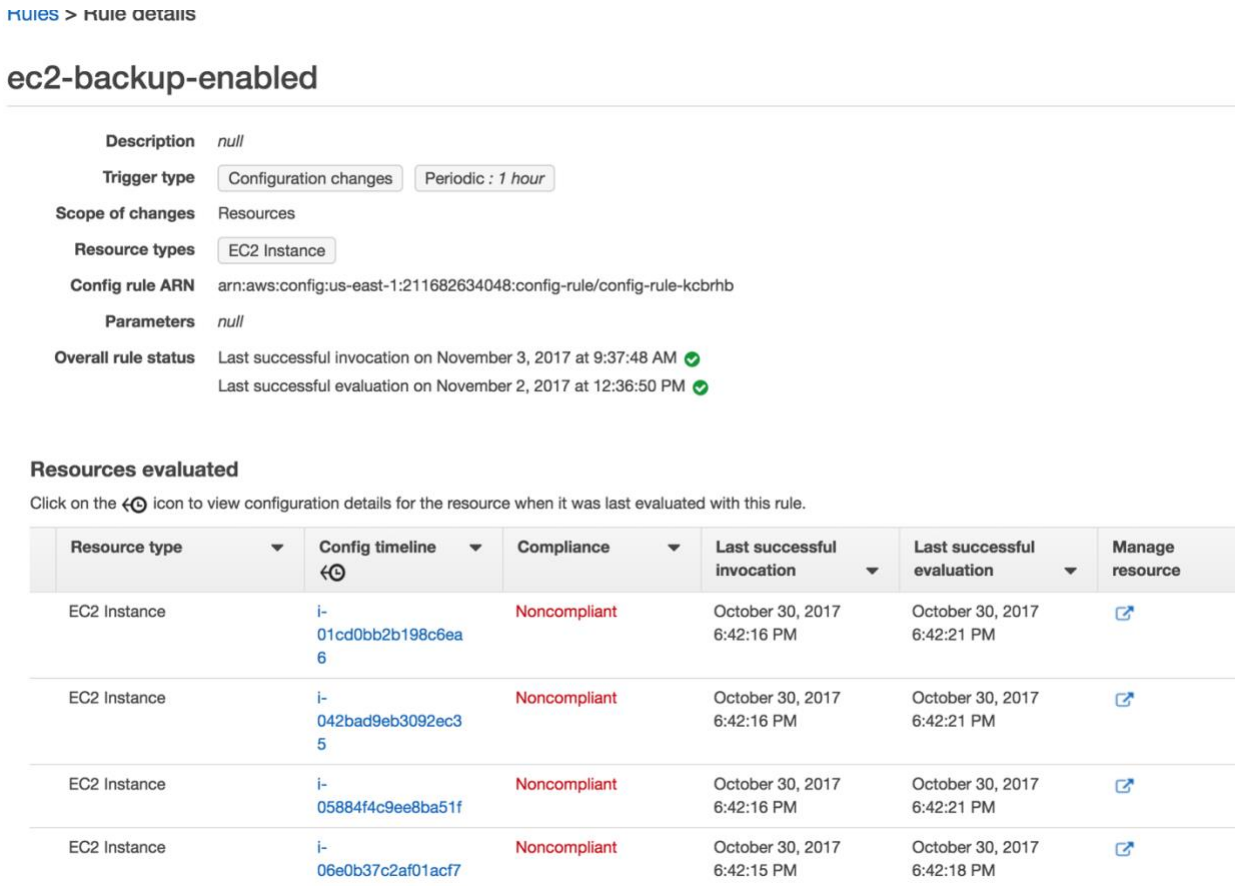
2. Click **Dashboard** to view the **Config Dashboard**.



3. Click **Rules** to view a list of available rules.



The **ec2-backup-enabled** rule should show a list of compliant and non-compliant EC2 instances.



SSM Agents

When Linux and Windows EC2 instances are created by launching simple workload stacks, AWS SSM agent processes are installed and started on them, at the latest version of the agent available from AWS at the time the workload was created.

AWS occasionally updates the SSM agents they provide. If you want to update the version of the SSM agent on all running workload instances for a region, follow these steps:

1. Download the following script from the S3 bucket containing the assets you used to launch the master stack for your account, to a system where you have the AWS CLI installed:

   ```
   deploy/utilities/updateSSMAgent.sh
   ```

2. Edit the file. At the bottom of the file, set the REGION_ARRAY to the list of regions you wish to scan for instances, such as:

   ```
   REGION_ARRAY=(us-east-1 ca-central-1 ap-south-1)
   ```

3. Run the script. It will scan all Linux and Windows instances in the defined region.

- For any Linux instance where the version of the SSM agent is earlier than the latest version available from AWS, it will update the agent.
- For all Windows instances, it will update the SSM agent to the latest version regardless of the existing version. The mechanism of updating the SSM agent on Windows does not provide the ability to determine if the agent is at the latest version. If the agent is already at the latest version, no agent changes will be made.

You should see an output like the following example.

In this case, only the i-00bbac29ec21f00dd instance was changed. If this was a Linux instance, the agent version was updated. If this was a Windows instance, the agent was updated if it was not at the latest version. If it was at the latest version, you will still see the same output. You will never see the output "... is already at the latest SSM Agent level" for a Windows instance.

```
# ./updateSSMAgent.sh
Working on instances in REGION us-west-1

Instance i-0f3440e48afd9859a is already at the latest SSM Agent
level
i-00bbac29ec21f00dd SSM agent not the latest so updating the
agent with command
Instance i-06cb705b25c5e2a39 is already at the latest SSM Agent
level
Instance i-0aefc0eaa17bb5a05 is already at the latest SSM Agent
level
Instance i-07a484e7fd2177b39 is already at the latest SSM Agent
level
Waiting on commands to complete (wait up to 5 min)
command completed 287dd8e5-471f-4e1d-9b96-77fc30da72f2 with
status Success
final command statuses
Command id 287dd8e5-471f-4e1d-9b96-77fc30da72f2 completed for
instance i-00bbac29ec21f00dd with status Success
```

Accessing Logs

16

Capturing Windows and Linux system logs is provided as a part of the base offering but is not considered a Gold Managed Service. The following chapter details how to access the logs that are created when instances are provisioned using the AMIs the Offering Build Team has provided. No manual configuration is needed.

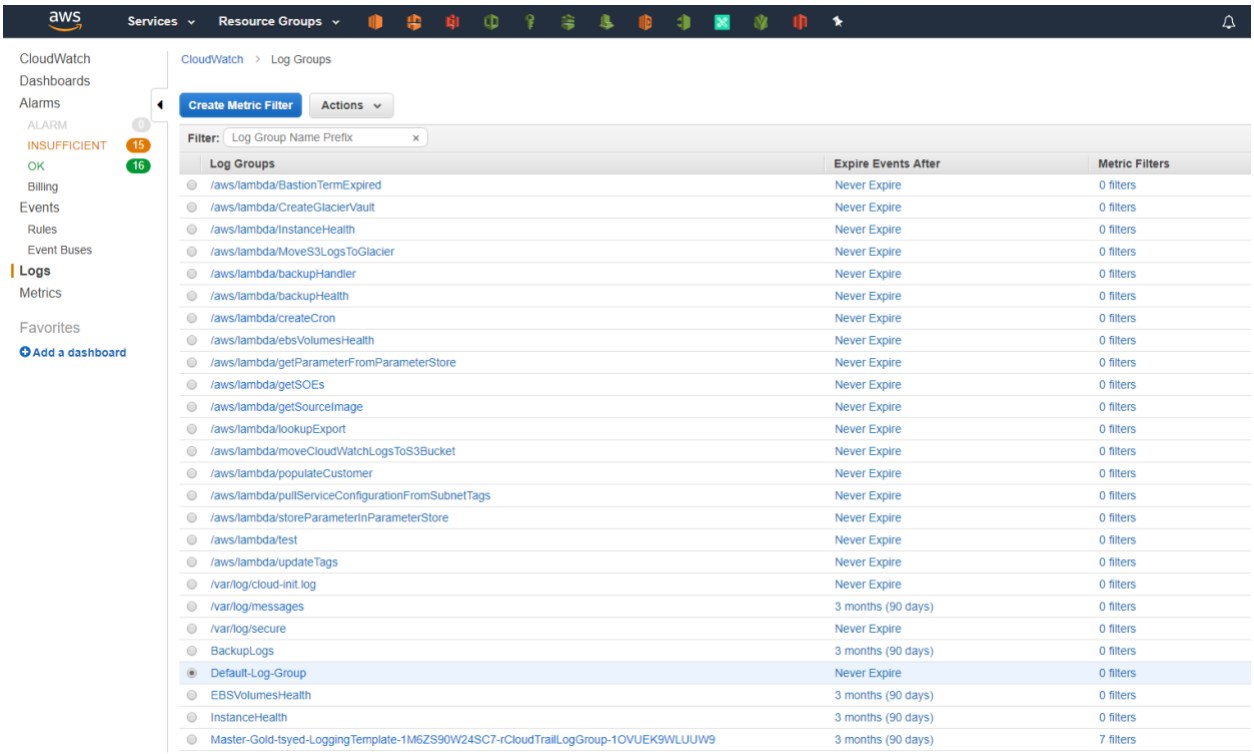
Viewing Windows Syslogs

Types of event logs in Windows:

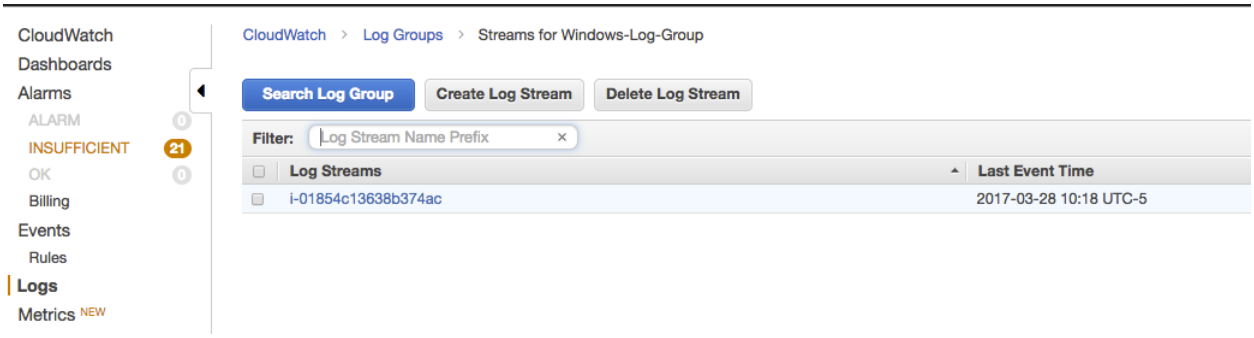
- System logs
- Application logs
- Security logs

To view Windows syslogs:

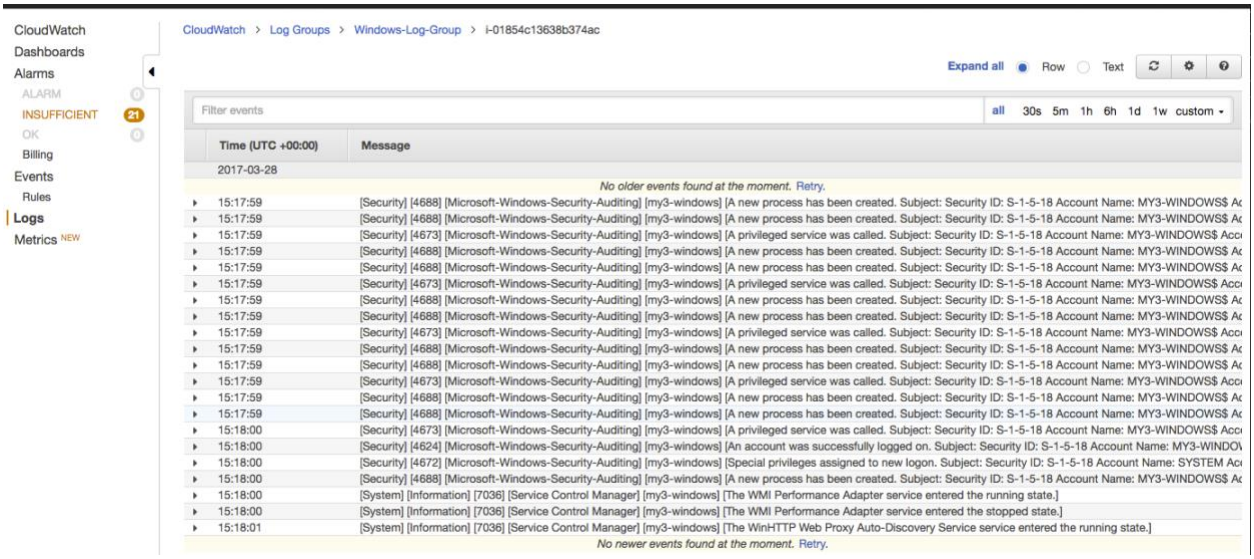
1. From the AWS console, navigate to **CloudWatch** > **Logs** > **Default-Log-Group**.



2. Click on the Instance ID to view the logs for your instance.



3. Look for the prefix [System], [Application], or [Security] to view System logs, Application logs, or Security logs in detail.

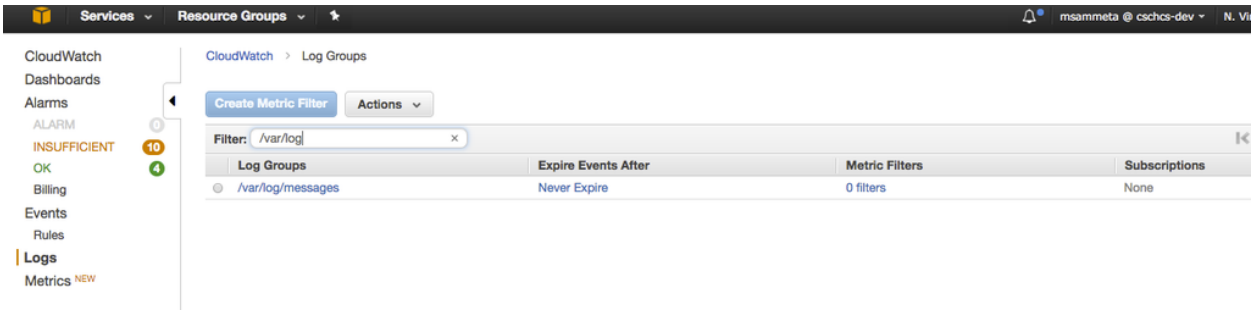


Viewing Red Hat Linux Syslogs

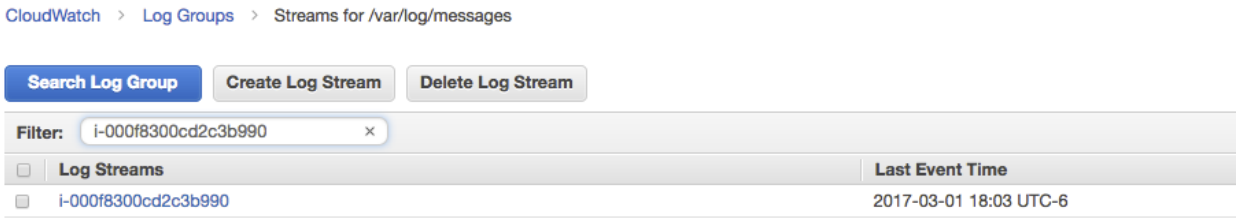
The CloudWatch agent captures the log files in the */var/log/messages* directory.

To view Red Hat syslogs:

1. Verify that the CloudWatch Log group is created successfully.
   CloudWatch agent is installed and configured as the instance starts up and a log group */var/log/messages* are automatically created as part of a Python script in Chef recipe awslogs-agent-setup.py.
2. **Go to Services** > **CloudWatch** > **Logs** > **/var/log/messages**.

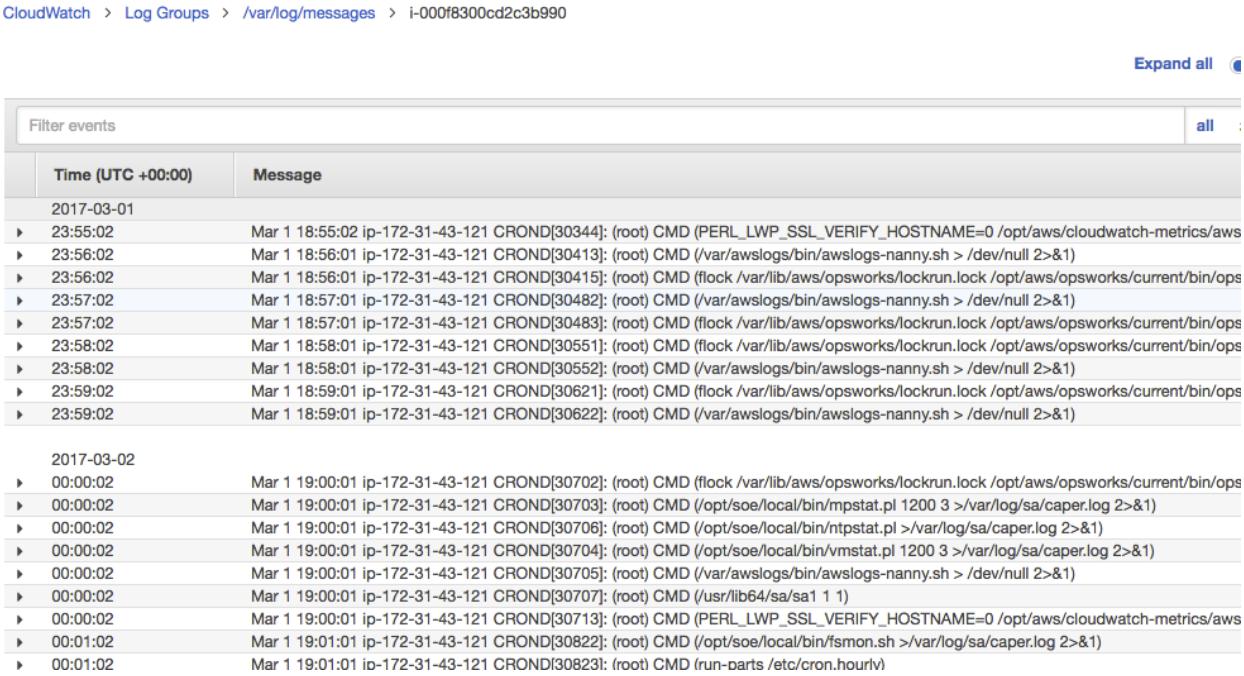


3. In the **Filter** field, paste the EC2 Instance ID and press Enter.



4. Click the Instance ID.

The log stream for the instances should be displayed



Use Amazon CloudWatch to monitor log files, set alarms, and react to changes.

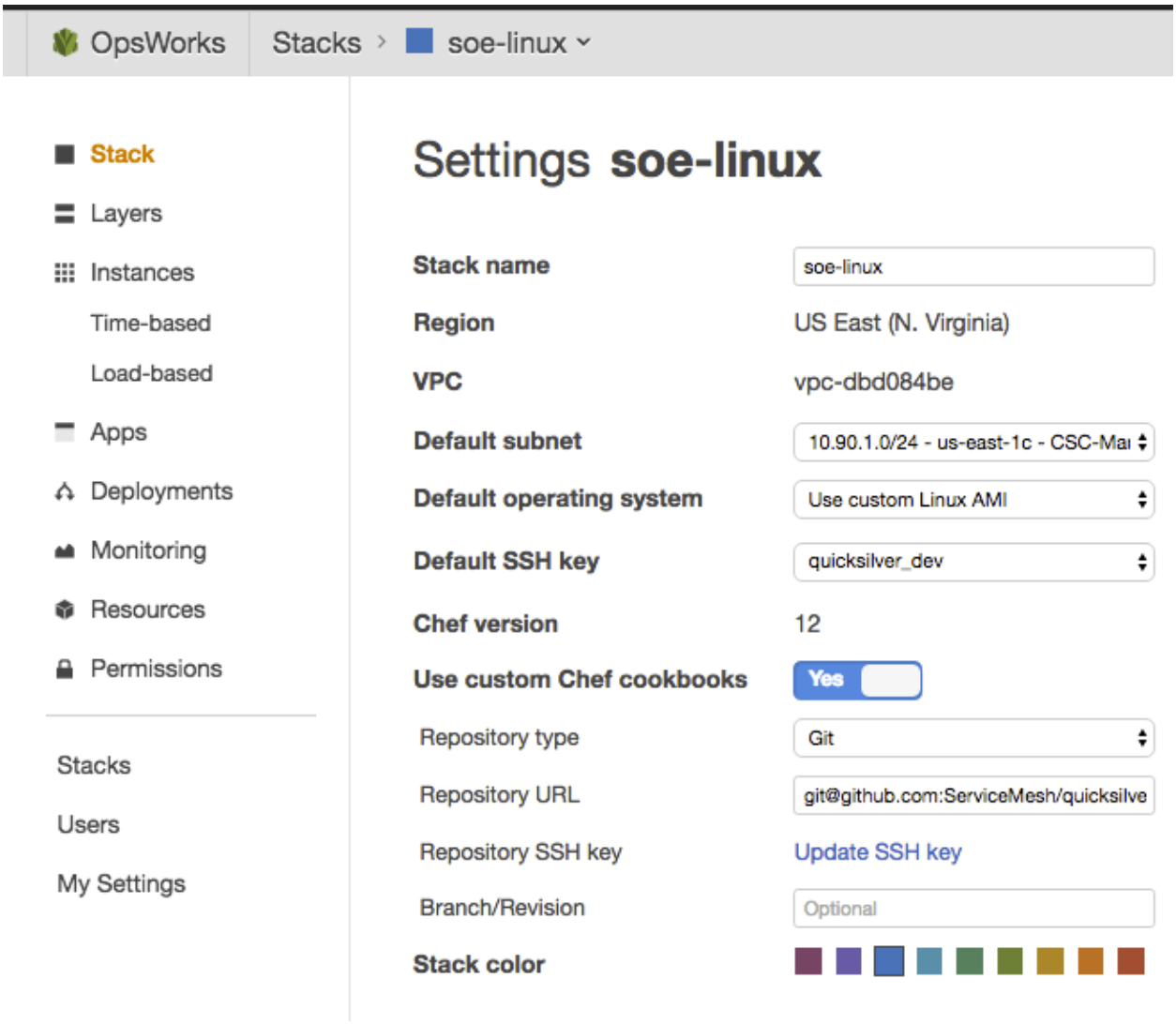
Components

Audit log assurance is achieved using a Chef recipe that:

- Installs and configures a CloudWatch Agent
- Captures Custom Metrics

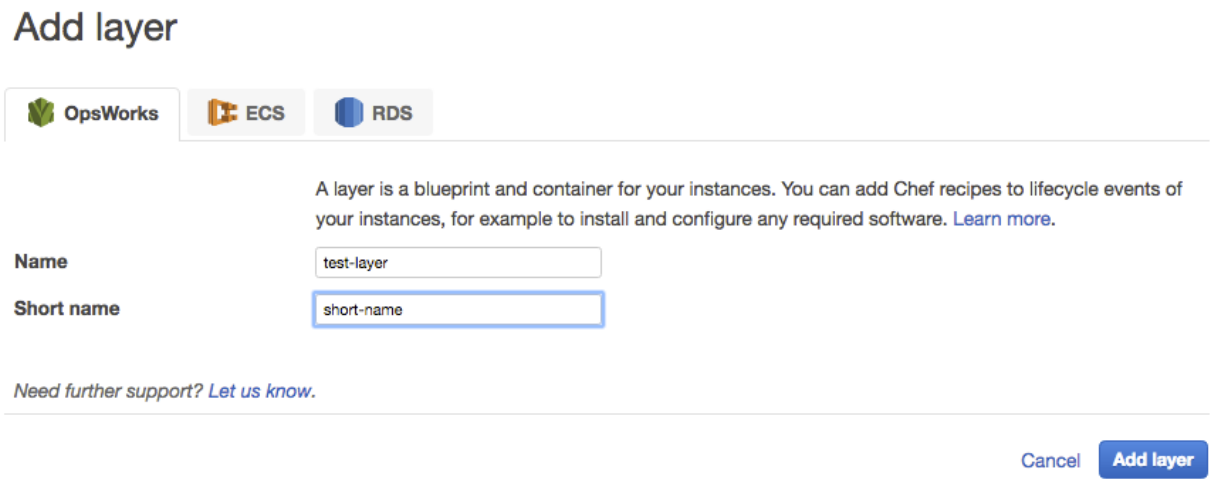
To enable Linux syslogs:

1. Create a new OpsWorks stack by navigating to **AWS** > **OpsWorks** > **Add Stack**.

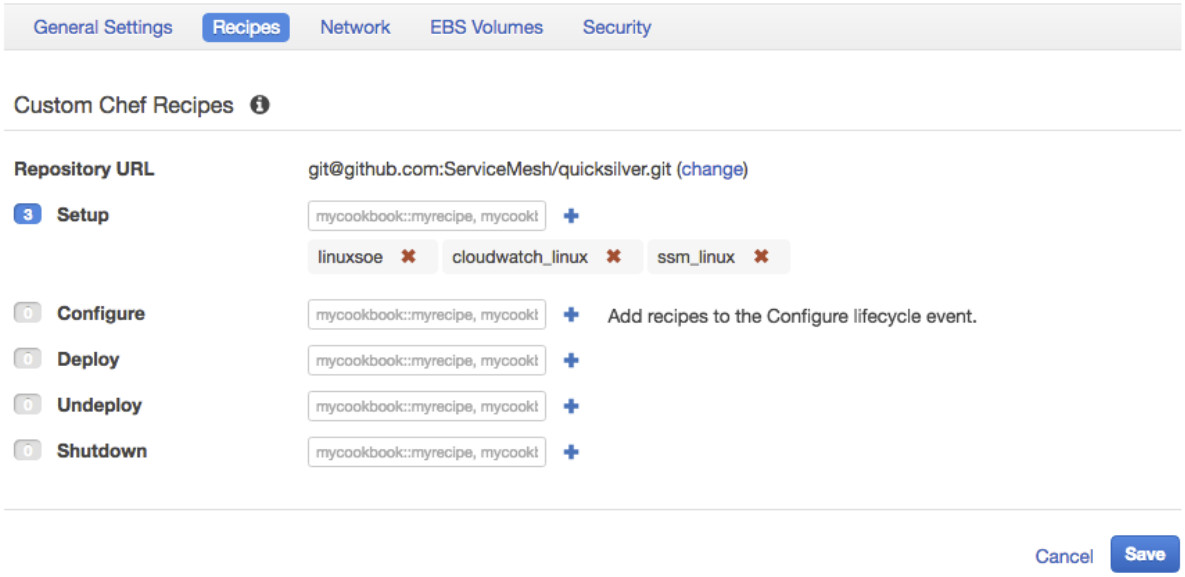


2. Create a new layer with the following Chef recipes:
   - *linuxsoe* (SOE Hardening)
   - *cloudwatch_linux (*installs, configures the CloudWatch agent and captures custom metrics*)*
3. Click **Add layer** on the confirmation that appears after creating the stack6.

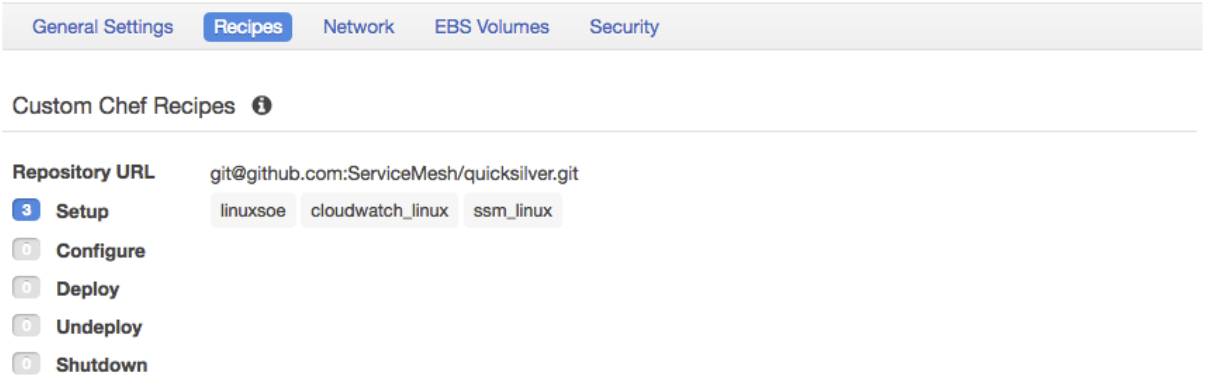
4. Enter a layer name.
5. Enter a layer short name.
6. Click **Add Layer**.



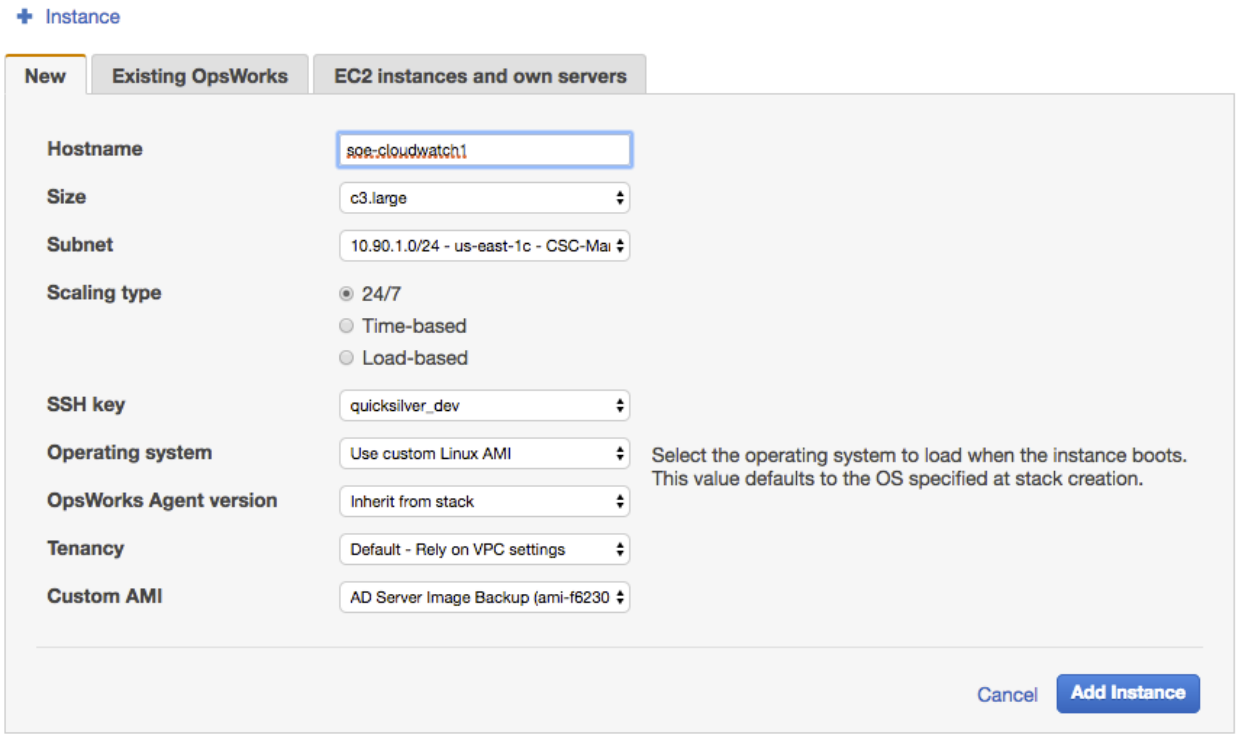
7. Click **Recipes** under the layer name.
8. Enter the recipe name **linuxsoe** next to Setup and click +.
9. Enter the recipe name **cloudwatch_linux** next to Setup and click +.



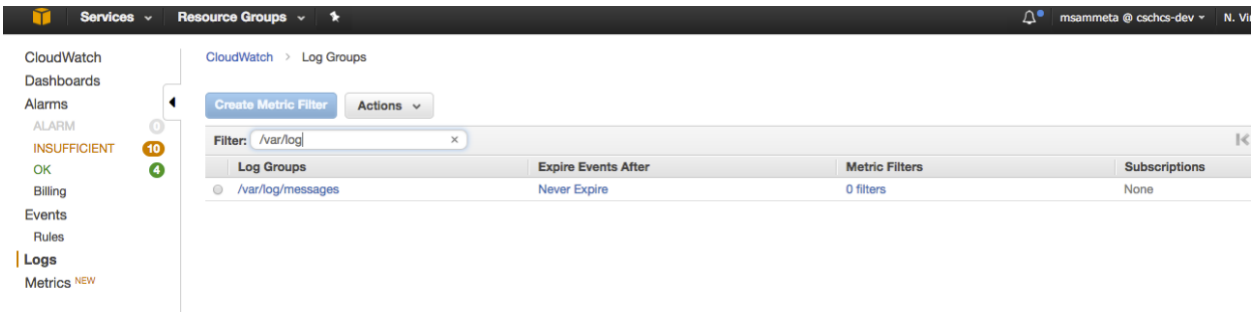
10. Click **Save** and the layer should be created.



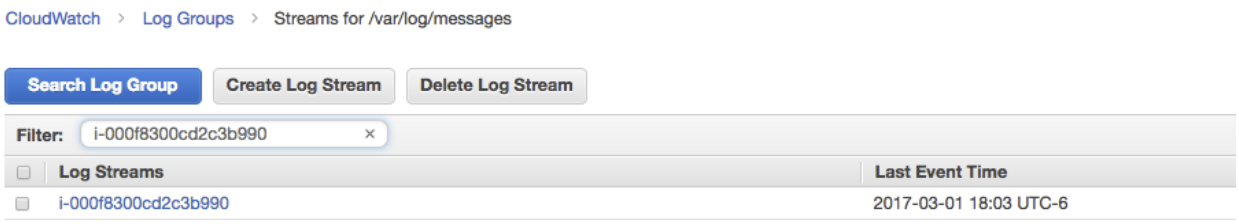
11. Create a new instance using custom AMIs.
    - `RHEL 72 - ami-6dcdd87a ( SOE-RHEL-7.2_HVM-20161025-x86_64 )`
    - `RHEL 67 - ami-ee2201f9 ( SOE-RHEL-6.7_HVM-20160412-x86_64 )`
12. Click **Instances**.
13. Click **Add an Instance**.
14. Enter an instance name.
15. Select the VM size.
16. Select the appropriate subnet.
17. Select the appropriate AMI name from the values above for the value of Custom AMI.
18. Click **Add Instance**.
19. Click **Start** and the instance should begin startup and configuration.
20. Once the instance completes startup, click the instance name if there are no errors.
21. Copy the value of EC2 Instance ID.
    This is the value used to identify the data in CloudWatch.
    Assumption: The instance is created with an IAM role that has a *CloudWatchFullAccess* policy attached to it.



22. Verify that the CloudWatch Log group is created successfully.
    CloudWatch agent is installed and configured as the instance starts up and a log group */var/log/messages* are automatically created as part of a python script in Chef recipe awslogs-agent-setup.py.
23. **Go to Services** > **CloudWatch** > **Logs** > **/var/log/messages**.

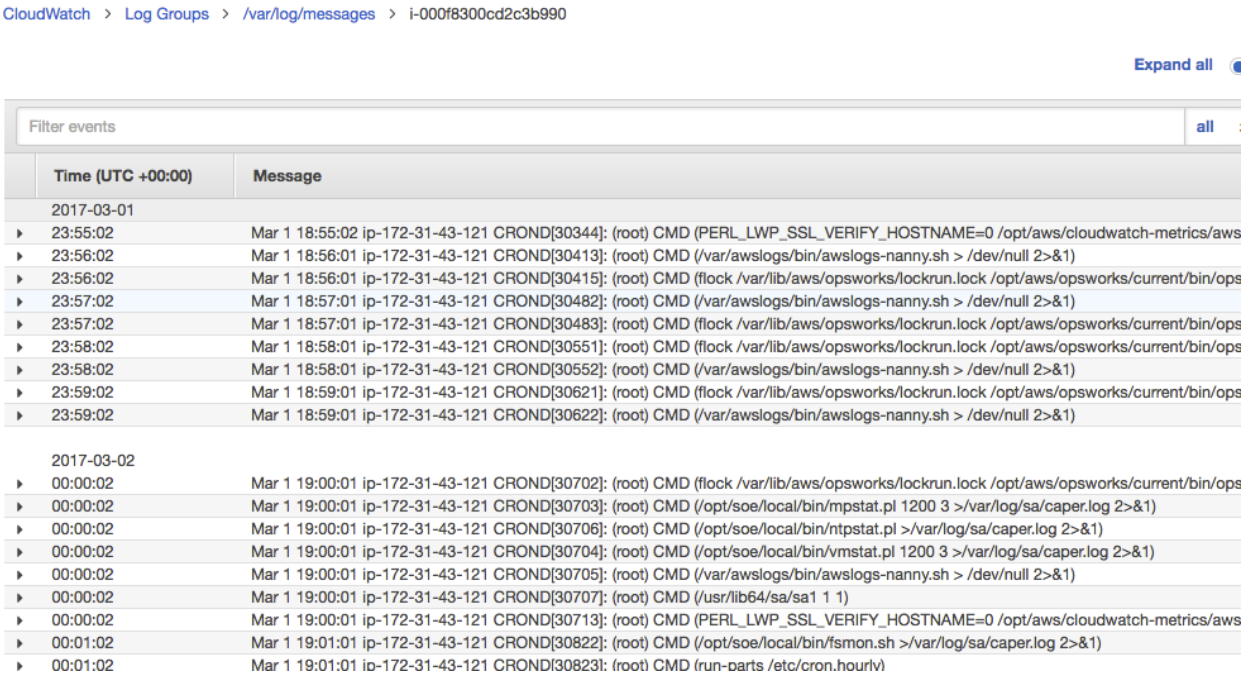


24. In the **Filter** field, paste the EC2 Instance ID and press Enter.



25. Click the Instance ID.

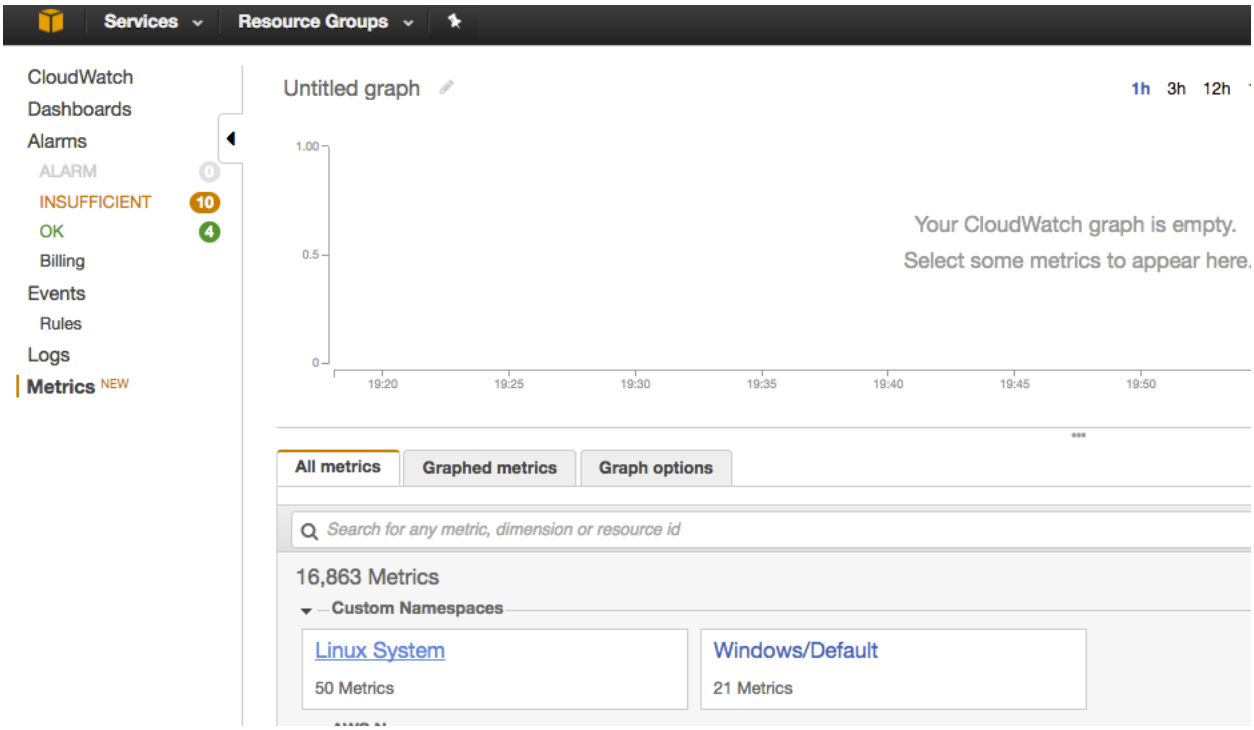
The log stream for Instance should be displayed

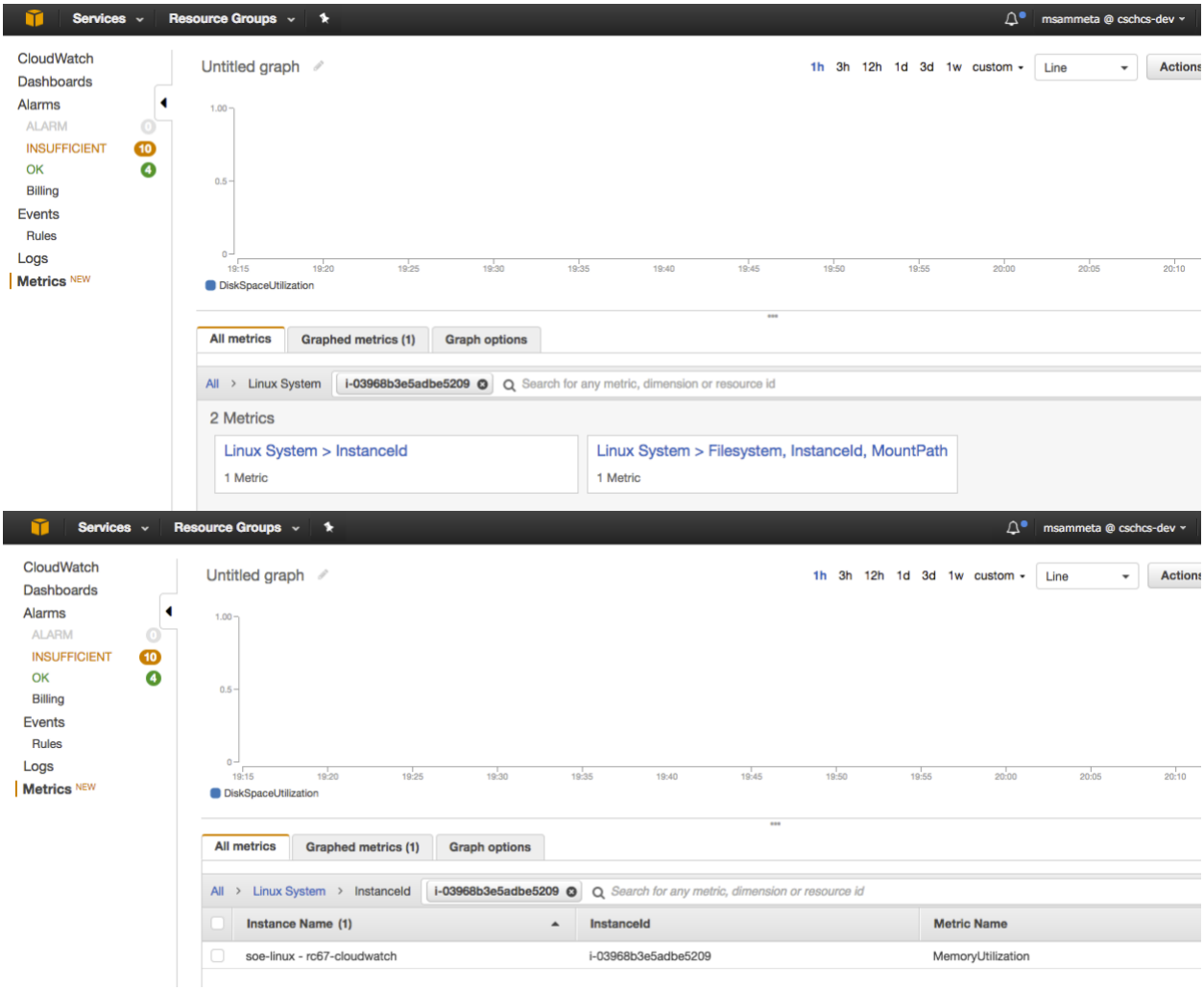


26. Verify that the CloudWatch Agent is installed and configured successfully.
27. SSH to the instance and verify that the *awslogs* daemon is running.
    This service is responsible for pushing logs to the log stream */var/log/messages*.

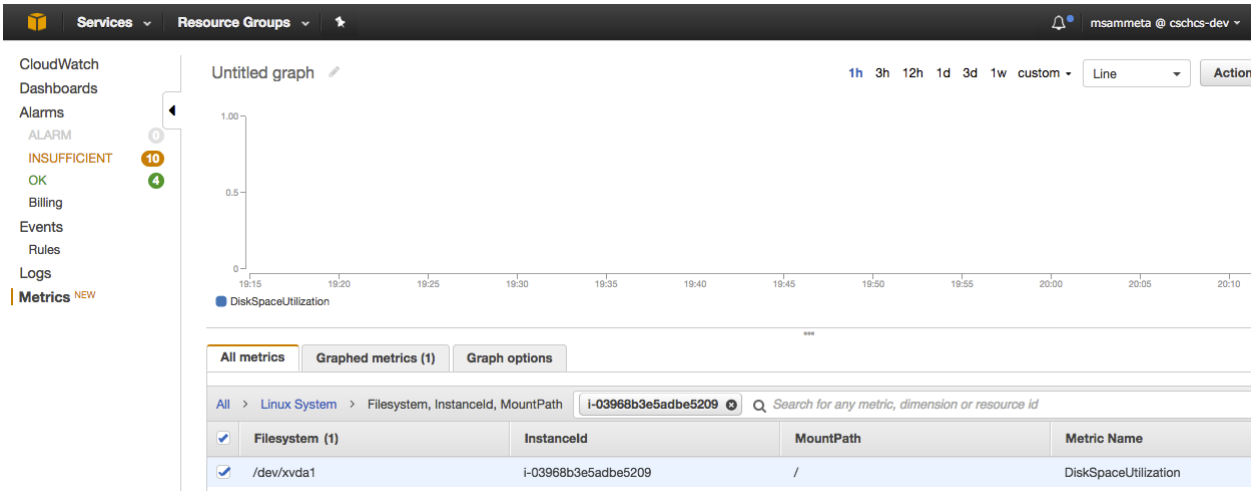
```
[ec2-user@ip-173-30-1-69 ~]$ sudo service awslogs status
• awslogs.service - LSB: Daemon for AWSLogs agent.
   Loaded: loaded (/etc/rc.d/init.d/awslogs; bad; vendor preset: disabled)
   Active: active (running) since Thu 2017-03-16 13:53:40 EDT; 2h 23min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 19764 ExecStop=/etc/rc.d/init.d/awslogs stop (code=exited, status=0/SUCCESS)
  Process: 19786 ExecStart=/etc/rc.d/init.d/awslogs start (code=exited, status=0/SUCCESS)
```

28. Verify that the Custom Metrics are configured successfully.
    The new instance that is created should report custom metrics *DiskSpaceUtilization* and *MemoryUtilization* under
    Services > **CloudWatch** > **Metrics** > **All Metrics** > **Custom Namespaces** > **Linux System** > **Filesystem, InstanceId, MountPath**
    Services > **CloudWatch** > **Metrics** > **All Metrics** > **Custom Namespaces** > **Linux System** > **InstanceId**





29. Select the metric that has the Instance ID and metrics should be displayed.



Viewing SUSE Logs and Metrics

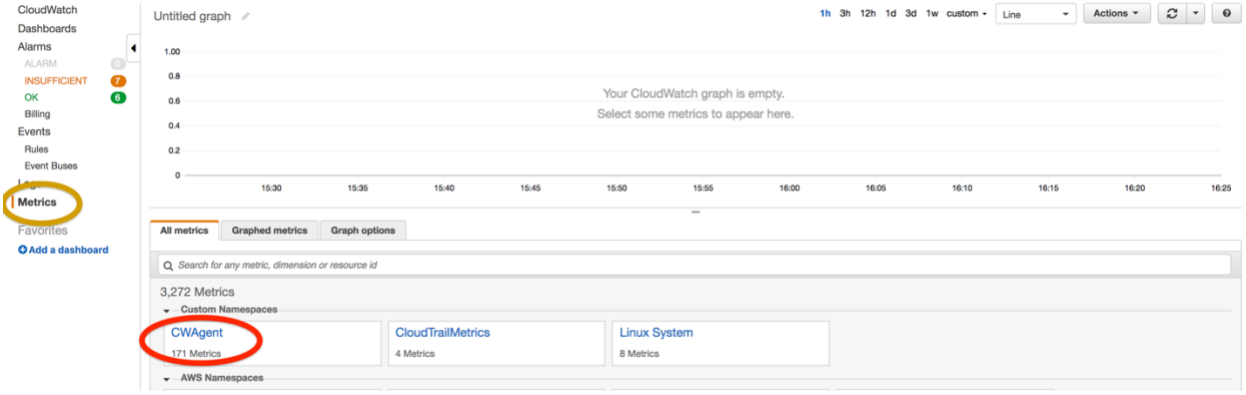
SUSE Linux Enterprise Server (SLES) 12 SP3 AMIs uses unified CloudWatch agent instead of legacy CloudWatch perl script. Unified CloudWatch agent uses SSM to update metrics and logs.

Metrics

The default Namespace for custom metrics is **CWAgent**.

To view the custom metrics for SLES AMIs:

1. From CloudWatch console, select **Metrics**.



2. Click **CWAgent** from the **All metrics** tab and custom metrics should be available.

Logs

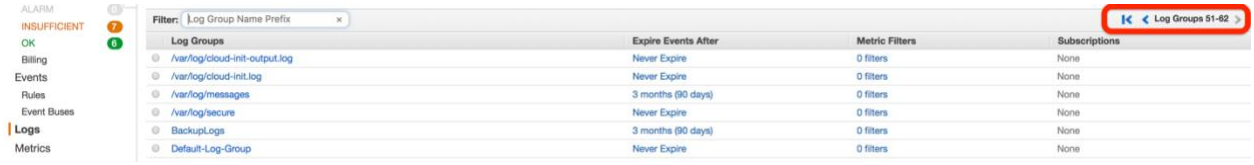
SLES 12 SP3 instances collects the following logs and creates log groups for them:

/var/log/messages

/var/log/cloud-init.log

/var/log/cloud-init-output.log

To access the log groups, go to **CloudWatch Dashboard → Logs**.



The **Logs** dashboard is paginated, you may have to go to the next page to find the Log Groups.

Monitoring Instance Health

17

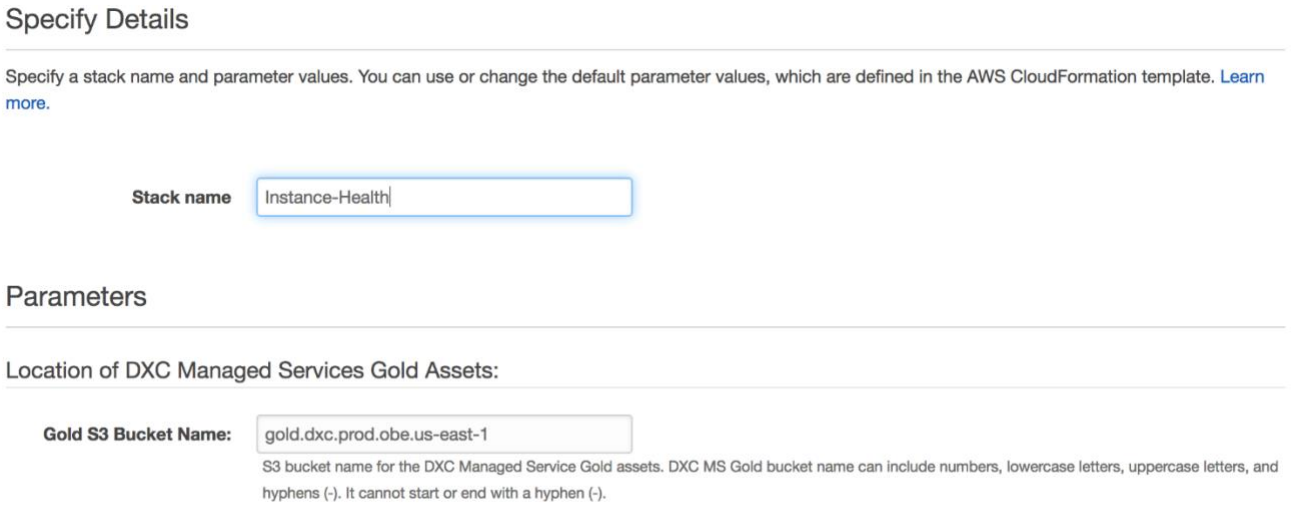
The Instance Health feature monitors various functions of a managed instance and reports errors when issues are found. The current functions monitored are:

- **Endpoint Protection** - Tests are made to ensure that the Endpoint Protection software is running on managed instances. Test verifies only that the software is running on the instances, it does not verify that the software is connected and reporting to the console. The second test will be added in a future release.
- **Logs** - Tests are made to verify that logs from the instance are being routinely stored in AWS CloudWatch. An alarm is generated if no new log entries have been made in the last 24 hours.

Installation

The Instance Health feature is installed as part of the Master Template process. To manually install the Instance Health feature, you must add it through the I**nstanceHealth.yam**l CloudFormation template. The only parameter is the location of the DXC Gold assets S3 bucket. Normally this will be in the standard location **gold.dxc.prod.obe.<region>** or it can be accessed from the customer bucket named **dxc.customer.config-<account-number>-<region>**.

The example shows adding the Instance Health feature to the current account in the us-east-1 region.



Installing the Instance Health feature does the following:

- Creates an IAM Role and a Lambda function to perform the health tests.
- Creates a Log Group named **InstanceHealth** and a Log Stream named **InstanceHealthMessages** to record issues in CloudWatch.
- Creates an SNS Topic named **InstanceHealthMessageTopic** for Alerts that are sent out from the health tests. The original email registered with the Customer Logging setup is registered as a subscription to this topic. If desired, a subscription to ServiceNow must be added manually.
- Creates a CloudWatch Rule to execute the Instance Health Lambda every 30 minutes at 0 and 30 minutes each hour. You can modify this schedule per customer requirements.

Processing

Running the instance health lambda does the following:

- The DXC instances are retrieved by looking at the tags for each instance. If the following four tags are found, the instance is considered to be a DXC instance that should be tested for health: *InstanceName*, *Compliance*, *OSName*, and *LeaseExpirationDate*. These tags were added to Instances launched by the CloudFormation templates in release 1.0. For release 1.1, the tag *DXCProduct* with the value *Quicksilver* was added. If this tag/value is found, then regardless of the four other tags, the instance is considered to be an instance that should be tested for health.
- The SSM Managed instances are retrieved. These instances registered with AWS through the SSM agent installed on the instance.
- Verifications are performed to further filter the instances:
  1. The state is checked. If the state is not *running*, the instance is skipped.
  2. The create time of the instance is checked. If the instance was launched in the last 15 minutes, it is skipped until it has been more than 15 since the launch time.
  3. The instance is checked to see if it is in the list of Managed instances. If it is not in the managed list, the SSM agent has not registered. This is a health issue so an Event and a log are generated.
  4. The SSM Ping Status status is checked. If the status is not *Online*, this is a health issue and an Event and log message are generated.
- The CrowdStrike status is checked by an EC2 Run Command executing a shell script on the instance looking for the Falcon Agent process. On Linux, this is a shell script looking for a process named */opt/CrowdStrike/falcon-sensor*. On Windows, the process name is either *CSNest* or *CSFalcon* (more likely *CSFalcon*). If this process cannot be found, this is a health issue and an Event and log message are generated.
- The CloudWatch logs are checked to see if the instance has generated a message in the last 24 hours. For Linux, the log group */var/log/messages* is checked. For Windows, the log group *Default-Log-Group* is checked. The log stream is the Instance ID of the instance. If an event in the log stream that is less than 24 hours old cannot be found, this is a health issue and an Event and log message are generated.

Instances that pass these tests are considered *healthy*. This status is noted in the log file and no additional events (emails or alarms) are generated.

About Resource Tags

18

When the Master stack is created, it creates resources as a part of the stack. The resources are tagged to support the platform and to keep track of the resources owned by DXC Managed Services for AWS.

Bookkeeping Tags:

Tags that are used to keep track of the resources owned by DXC. e.g. `Owner: DXC`

Platform Tags:

Tags that are used by the AWS Managed Platform. e.g. `os: amazon-linux`

List of Tags

VPC

Services	Resources	Key	Values	Remarks
VPC	VPC	Name	Management VPC Customer Workload VPC	To identify the EC2 and S3 resources owned by DXC
		Owner	DXC	
	Route Table	Name		
		Network	Public Private	
		Owner	DXC	
		RouteTableType	Public Private	
	Subnet	Name	Company Name X	
		Network	Public Subnet	
		Owner	DXC	
		SubnetType	Public Private	

EC2

Services	Resources	Key	Values	Remarks
EC2	AMI	ami	quicksilver	
		os	win2016 amazon-linux rhel7.2 rhel6.7	
		osservicelevel	GOLD SILVERPLUS	
		encrypted	true false	
		Owner	DXC	
	Instance	Project	Quicksilver	
		Compliance		
		Backup	true false	
		BackupSchedule		
		RetentionPeriod	e.g. 30	
		ApplyPatching	true false	
		Patch Group		
		ApplyLogging	true false	
		ApplyMonitoring	true false	
		ApplyEndPointProtection	true false	
		BusinessUnit	e.g. Cloud	
		DXCProduct	Quicksilver	

	LeaseExpirationDate		
	Environment	e.g. QA	
	Name	e.g. Customer WL	
	InstanceName	e.g. Customer WL	
	Application		
	Owner		
	OSServiceLevel	GOLD SILVERPLUS	
	OSName		

S3

Services	Resources	Key	Values	Remarks
S3	Buckets	Owner	DXC	

Lambda

Services	Resources	Key	Values	Remarks
Lambda	Functions	Owner	DXC	