

## Module 5 Lab 2: Amazon EKS

The screenshot shows the AWS IAM Home page. On the left, a sidebar navigation includes 'Access management' (Groups, Users, Roles), 'Access reports' (Access analyzer, Archive rules, Analyzer details), and 'AWS account ID'. The main area displays 'IAM Resources' with counts for Users (3), Groups (3), Roles (21), and Identity Providers (0). Below this is a 'Customer Managed Policies' section with a count of 8. A 'Security Status' bar indicates 3 out of 5 complete, with items including 'Activate MFA on your root account' (warning icon), 'Create individual IAM users' (checkmark), 'Use groups to assign permissions' (checkmark), 'Apply an IAM password policy' (warning icon), and 'Rotate your access keys' (checkmark). To the right, there's a video player showing a thumbnail of a person speaking, and an 'Additional Information' section with links to IAM best practices, documentation, and videos.

### Starting at IAM select Roles

The screenshot shows the AWS IAM Roles page. The sidebar navigation is identical to the previous screen. The main content area is titled 'Create role' and 'Delete role'. It features a search bar and a table listing 21 existing roles. The columns are 'Role name', 'Trusted entities', and 'Last activity'. The table lists roles such as 'aws-elasticbeanstalk-ec2-role', 'aws-elasticbeanstalk-service-role', 'AWSCodePipelineServiceRole-us-west-2-cl-cd-pipeline', and various AWS Service-Linked roles like 'autoscaling', 'cloud9', 'dax', 'ecs', 'elasticloadbalancing', 'rds', 'support', and 'trustedadvisor'. The last column shows the last activity period for each role.

### Select Create role

Service	Amazon CloudWatch Metrics	Greengrass	OpsWorks	Storage Gateway
Batch	DataSync	GuardDuty	Personalize	Textract
Chime	DeepLens	Health Organizational View	QLDB	Transfer
CloudFormation	Directory Service	IAM Access Analyzer	RAM	Trusted Advisor
CloudHSM	DynamoDB	Inspector	RDS	VPC
CloudTrail	EC2	IoT	Redshift	WorkLink
CloudWatch Application Insights	EC2 - Fleet	IoT Things Graph	Rekognition	WorkMail
CloudWatch Events	<b>EKS</b>			
CodeBuild				

Select your use case

**EKS**  
Allows EKS to manage clusters on your behalf.

**EKS - Fargate pod**  
Allows access to other AWS service resources that are required to run Amazon EKS pods on AWS Fargate.

**EKS - Fargate profile**  
Allows EKS to run Fargate tasks.

**EKS - Nodegroup**  
Allow EKS to manage nodegroups on your behalf.

\* Required      Cancel      **Next: Permissions**

Feedback    English (US)      © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.    Privacy Policy    Terms of Use

## Selecting EKS as the use case

Create role

Attached permissions policies

The type of role that you selected requires the following policy.

Policy name	Used as	Description
AmazonEKSClusterPolicy	None	This policy provides Kubernetes the permission...
AmazonEKSServicePolicy	None	This policy allows Amazon Elastic Container S...

Set permissions boundary

\* Required      Cancel      Previous      **Next: Tags**

Feedback    English (US)      © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.    Privacy Policy    Terms of Use

## Included default policies

## Create role

1 2 3 4

### Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text"/>	<input type="text"/>	<a href="#">Remove</a>

You can add 50 more tags.

[Cancel](#) [Previous](#) [Next: Review](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

## Optional tags

## Create role

1 2 3 4

### Review

Provide the required information below and review this role before you create it.

**Role name\***  Use alphanumeric and '+', '=', '@', '-' characters. Maximum 64 characters.

**Role description**  Maximum 1000 characters. Use alphanumeric and '+', '=', '@', '-' characters.

**Trusted entities** AWS service: eks.amazonaws.com

**Policies**  [AmazonEKSClusterPolicy](#)  [AmazonEKSServicePolicy](#)

**Permissions boundary** Permissions boundary is not set

No tags were added.

\* Required

[Cancel](#) [Previous](#) [Create role](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

## Review before creating

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with various options like Dashboard, Access management, Roles, and Access reports. The 'Roles' section is currently selected. In the main content area, there's a search bar at the top with the query 'eks'. Below it is a table with three columns: 'Role name', 'Trusted entities', and 'Last activity'. A single row is visible, showing 'eks-role' under 'Role name', 'AWS service: eks' under 'Trusted entities', and 'None' under 'Last activity'. At the bottom of the page, there's a message 'Role created'.

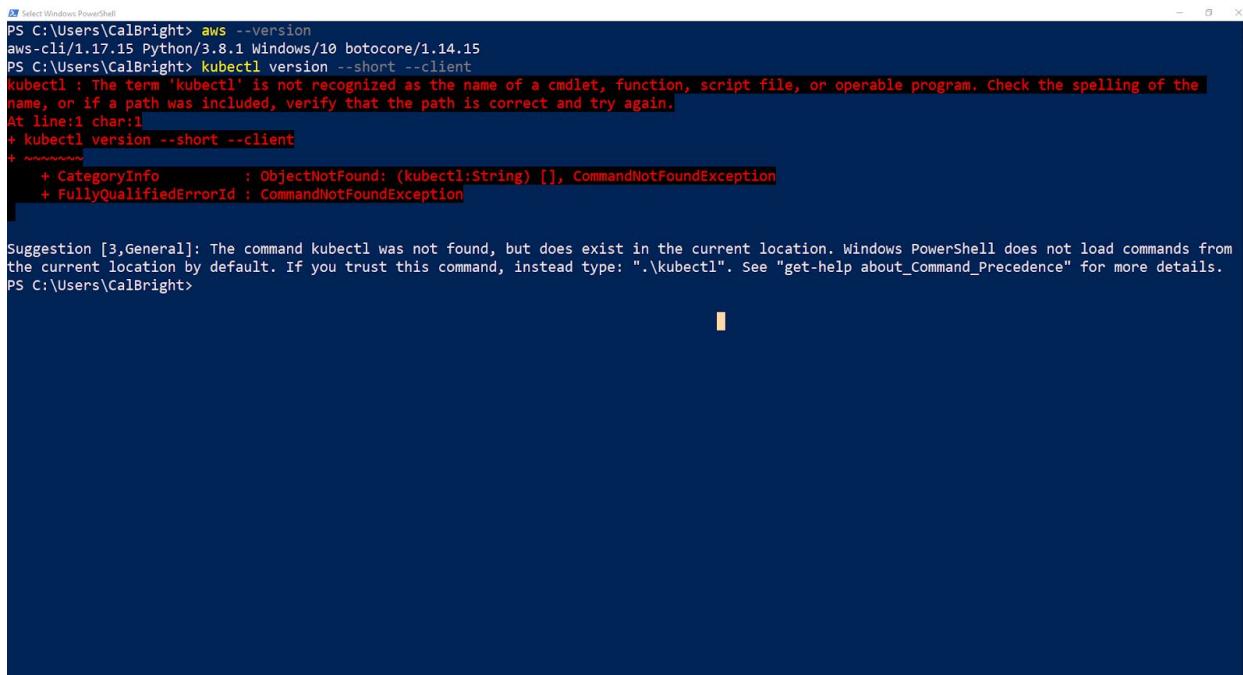
The screenshot shows a Windows PowerShell window. The command run is:

```
PS C:\Users\CalBright> Get-ChildItem -Path C:\ -Include *kubectl* -File -Recurse -ErrorAction SilentlyContinue | Where-Object { $_.LastWriteTime -ge $FindDate }
```

The output shows a single file named 'kubectl' located in the 'C:\Users\CalBright' directory, with its LastWriteTime being 3/13/2020 6:45 PM and Length being 48633128 bytes.

Kubectl has downloaded to the local folder. FindDate set then used with Get-ChildItem.  
“\$FindDate=Get-Date -Year 2020 -Month 03 -Day 13”

“Get-ChildItem -Path C:\ -Include \*kubectl\* -File -Recurse -ErrorAction SilentlyContinue | Where-Object { \$\_.LastWriteTime -ge \$FindDate }”



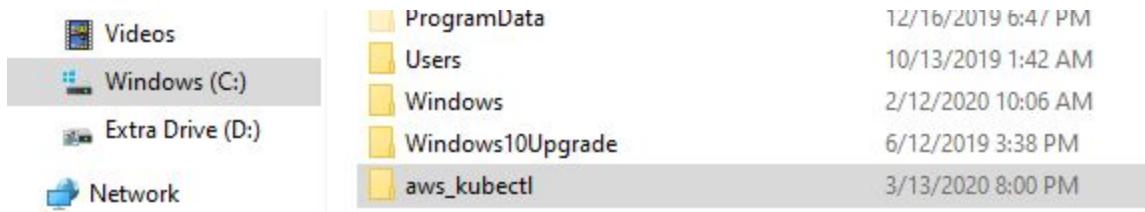
```
PS C:\Users\CalBright> aws --version
aws-cli/1.17.15 Python/3.8.1 Windows/10 botocore/1.14.15
PS C:\Users\CalBright> kubectl version --short --client
kubectl : The term 'kubectl' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ kubectl version --short --client
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (kubectl:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Suggestion [3,General]: The command kubectl was not found, but does exist in the current location. Windows PowerShell does not load commands from
the current location by default. If you trust this command, instead type: ".\kubectl". See "get-help about_Command_Precedence" for more details.
PS C:\Users\CalBright>
```

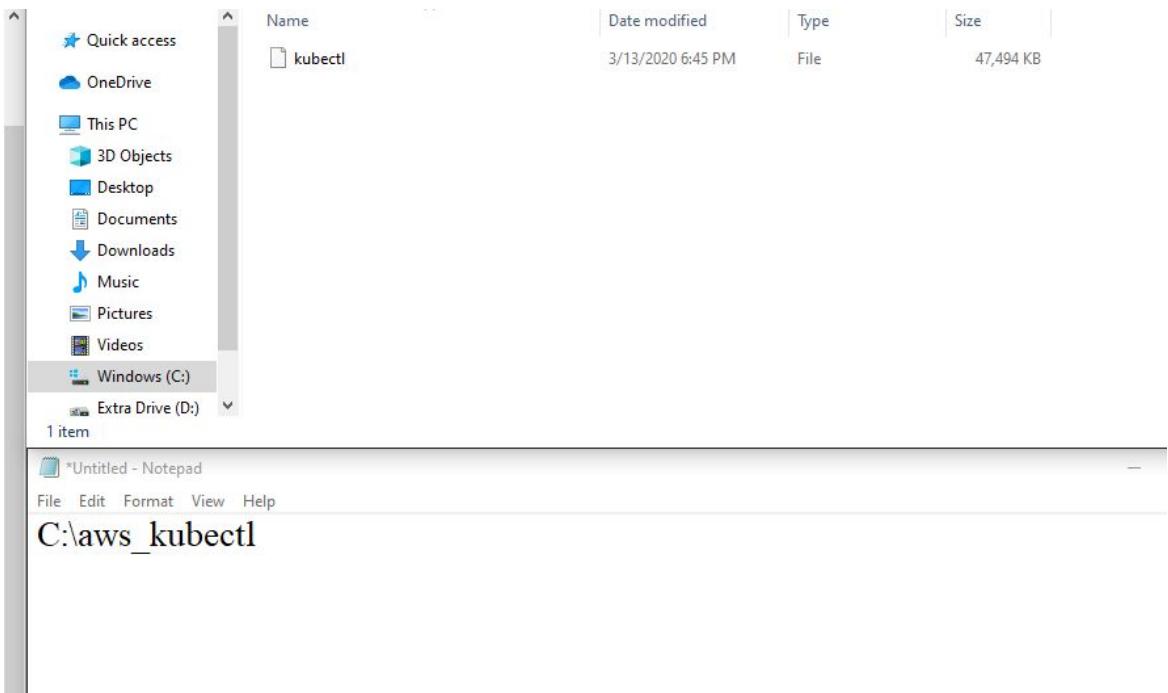
Without any edits to the system path, Powershell warns me of its location. Note that aws is present but Kubectl is not.

.packettracer	11/2/2019 1:00 PM	PACKETTRACER File	1 KB
EC2KeyPair.pem	2/23/2020 11:30 PM	PEM File	2 KB
<b>kubectl</b>	<b>3/13/2020 6:45 PM</b>	<b>File</b>	<b>47,494 KB</b>
NTUSER.DAT	3/13/2020 2:49 PM	DAT File	6,144 KB

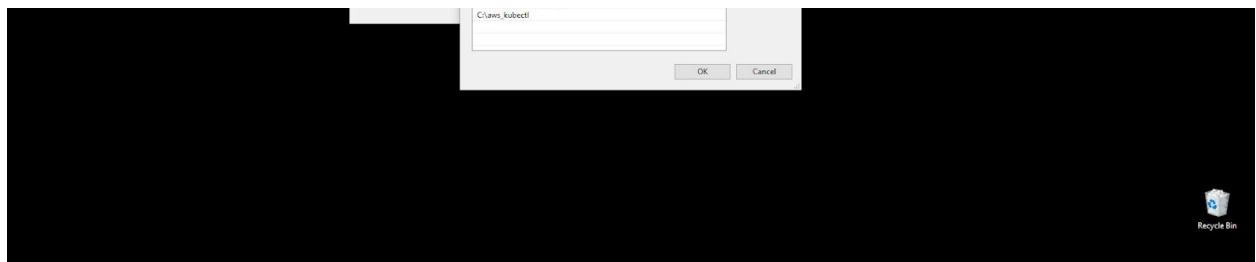
First I'll take the downloaded kubectl and copy it in a new folder on my local drive



New folder where kubectl will be placed



Location of kubectl



Added C:\aws\_kubectl to the environment variable list as a new entry.



Documentation Blog Training Partners Community Case Studies English ^ v1.17 ^

2. Test to ensure the version you installed is up-to-date:

```
kubectl version --client
```

## Install kubectl on Windows

Install kubectl binary with curl on Windows [🔗](#)

1. Download the latest release v1.17.0 from [this link](#).

Or if you have `curl` installed, use this command:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.17.0/bin/windows/amd64/kubectl.exe
```

To find out the latest stable version (for example, for scripting), take a look at <https://storage.googleapis.com/kubernetes-release/release/stable.txt>.

2. Add the binary to your PATH.

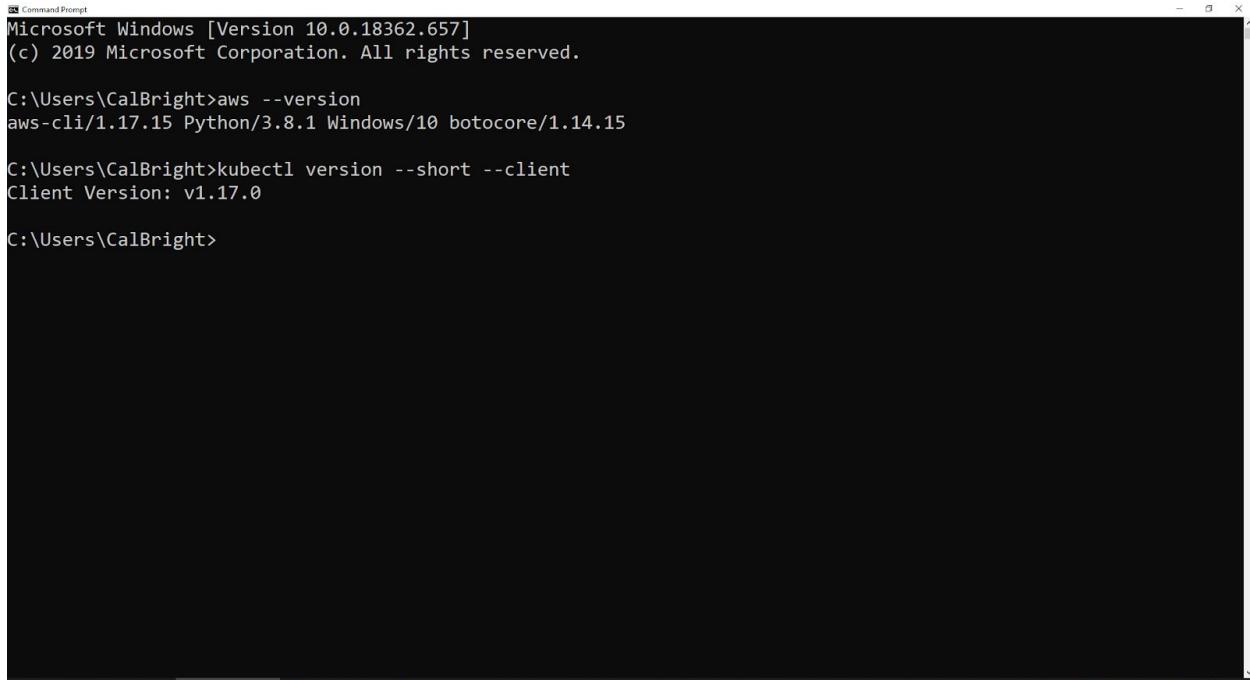
3. Test the version of `kubectl` is the same as downloaded:

```
kubectl version --client
```

Before continuing, I downloaded the latest kubectl using the kubernetes website



Placing in with the older version



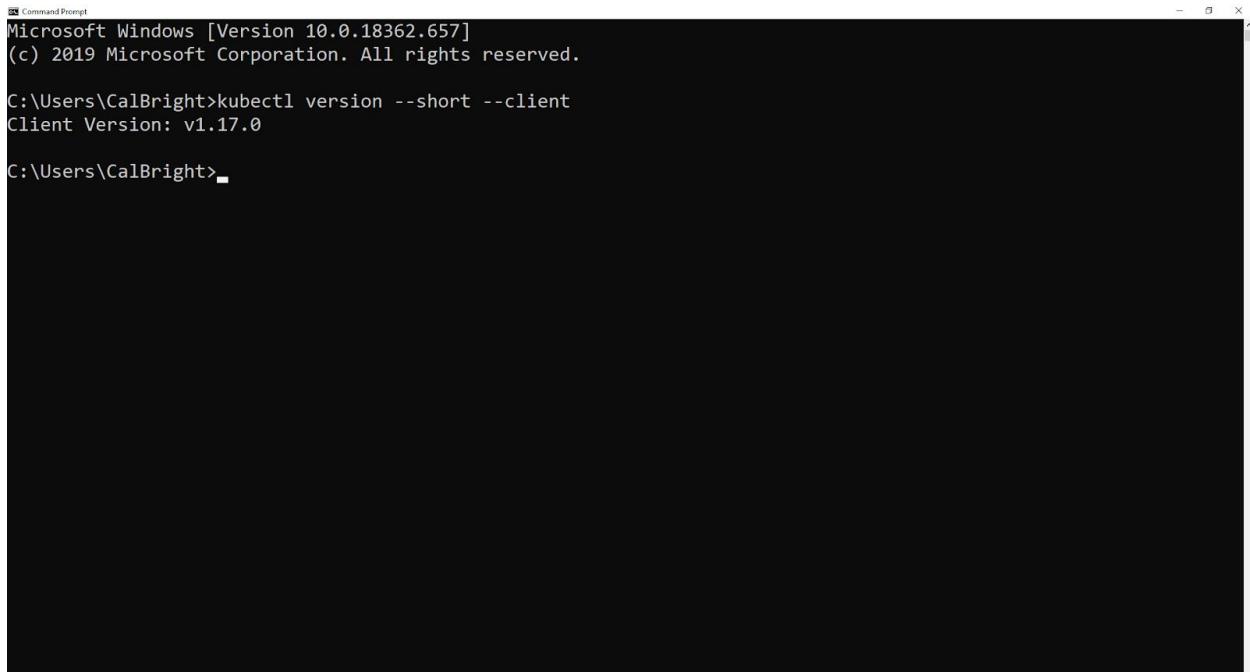
```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\CalBright>aws --version
aws-cli/1.17.15 Python/3.8.1 Windows/10 botocore/1.14.15

C:\Users\CalBright>kubectl version --short --client
Client Version: v1.17.0

C:\Users\CalBright>
```

Note that client version for kubectl



```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\CalBright>kubectl version --short --client
Client Version: v1.17.0

C:\Users\CalBright>
```

Removing the older file has no effect on version output

The screenshot shows the AWS CloudFormation Stacks page. On the left, there's a sidebar with navigation links like 'Stacks', 'StackSets', 'Exports', 'Designer', and sections for 'CloudFormation registry' and 'Resource types'. Below the sidebar are links for 'Previous console' and 'Feedback'. The main content area has a header 'CloudFormation > Stacks' and a sub-header 'Stacks (0)'. It features a search bar with a placeholder 'Filter by stack name' and a dropdown menu with 'Active' selected. There are buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'. A table below the header shows columns for 'Stack name', 'Status', 'Created time', and 'Description'. A message 'No stacks' and 'No stacks to display' is centered. At the bottom is a large orange 'Create stack' button and a link 'View getting started guide'.

### From the CloudFormation Create Stack (standard)

The screenshot shows the 'Create stack' wizard, Step 1: Specify template. On the left, a sidebar lists steps: 'Step 1 Specify template', 'Step 2 Specify stack details', 'Step 3 Configure stack options', and 'Step 4 Review'. The main area has a title 'Create stack' and a sub-section 'Prerequisite - Prepare template'. It contains a note about templates being JSON or YAML files and three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. Below this is a 'Specify template' section with a note about templates describing stack resources. It has a 'Template source' section for selecting an Amazon S3 URL, with 'Amazon S3 URL' selected and a text input field containing 'https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon-eks-vpc-sample.yaml'. There's also an 'Upload a template file' option and a 'View in Designer' button. At the bottom are 'Cancel' and 'Next' buttons.

The template file<sup>1</sup> will facilitate the stack creation process

<sup>1</sup> <https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html>

We recommend a network architecture that uses private subnets for your worker nodes, and public subnets for Kubernetes to create public load balancers within.

Choose the tab below that represents your desired VPC configuration.

**Public and private subnets**      Only public subnets

**To create your cluster VPC with public and private subnets**

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>.
2. From the navigation bar, select a Region that supports Amazon EKS.
3. Choose **Create stack**.
4. For **Choose a template**, select **Specify an Amazon S3 template URL**.
5. Paste the following URL into the text area and choose **Next**:

```
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon
```

6. On the **Specify Details** page, fill out the parameters accordingly, and then choose **Next**.

Initial cluster created using the wrong template (public and private subnets instead of public subnets)

**Clusters**

Amazon ECR  
Repositories

**Node Groups (0) Info**

Group name	Desired size	AMI release version	Status
No Managed Node Groups			
This cluster does not have any Managed Node Groups.			
Nodes that are not part of an Amazon EKS Managed Node Group are not shown in the AWS console.			
<b>Add Node Group</b>			

**Networking**

VPC vpc-06742caefc8937cb1	Subnets subnet-04385894bcb01f7b7 subnet-01cd1af7553101956 subnet-02a3d8fb9784eecd8 subnet-068e87abf0866d384	Cluster security group sg-08cc81c7ebdb2e68f	API server endpoint access Private access Disabled  Public access Enabled  Public access source whitelist 0.0.0.0 (open to all traffic)
------------------------------	---	--	---

**Note the incorrect 4 subnets**

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with various VPC-related options like Subnets, Route Tables, Internet Gateways, etc. The main area displays two subnets:

- Subnet ID:** subnet-02a3d8fb9784ecc8
- Name:** eks-vpcpub-stack-PrivateSubnet02
- State:** available
- VPC:** vpc-06742cae0c8937c

On the right, another subnet is shown:

- Subnet ID:** subnet-04385894bcd01f7b7
- Name:** eks-vpcpub-stack-PublicSubnet01
- State:** available
- VPC:** vpc-06742cae0c8937c

Below the subnets, there are tabs for Description, Flow Logs, Route Table, Network ACL, Tags, and Sharing.

Note the Private on the left and Public on the right. Running a worker node may not work on a private subnet.

The screenshot shows the AWS Documentation for Amazon EKS User Guide. The page title is "Public and private subnets". It includes a note: "Only public subnets". Below this, there's a section titled "To create your cluster VPC with only public subnets" with the following steps:

- Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>.
- From the navigation bar, select a Region that supports Amazon EKS.
- Choose **Create stack**.
- For **Choose a template**, select **Specify an Amazon S3 template URL**.
- Paste the following URL into the text area and choose **Next**:

```
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon-eks-vpc-sample.yaml
```

Further steps are listed:

- On the **Specify Details** page, fill out the parameters accordingly, and then choose **Next**.
  - Stack name:** Choose a stack name for your AWS CloudFormation stack. For example, you can call it **eks-vpc**.
  - VpcBlock:** Choose a CIDR range for your VPC. You can keep the default value.
  - Subnet01Block:** Specify a CIDR range for subnet 1. We recommend that you keep the default value so that you have plenty of IP addresses for pods to use.
  - Subnet02Block:** Specify a CIDR range for subnet 2. We recommend that you keep the default value so that you have plenty of IP addresses for pods to use.
  - Subnet03Block:** Specify a CIDR range for subnet 3. We recommend that you keep the default value so that you have plenty of IP addresses for pods to use.
- (Optional) On the **Options** page, tag your stack resources. Choose **Next**.

The public subnets template file<sup>2</sup> will aid in creating the EKS vps

<sup>2</sup> <https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html>

Specify stack details

**Stack name**

Stack name: eks-vpc

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**Worker Network Configuration**

**VpcBlock**  
The CIDR range for the VPC. This should be a valid private (RFC 1918) CIDR range.  
192.168.0.0/16

**Subnet01Block**  
CidrBlock for subnet 01 within the VPC  
192.168.64.0/18

**Subnet02Block**  
CidrBlock for subnet 02 within the VPC  
192.168.128.0/18

**Subnet03Block**  
CidrBlock for subnet 03 within the VPC. This is used only if the region has more than 2 AZs.  
192.168.192.0/18

**Cancel** **Previous** **Next**

**Feedback** **English (US)**

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

## Stack named eks-vpc-stack

CloudFormation > Stacks > Create stack

Configure stack options

**Tags**  
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove
<a href="#">Add tag</a>		

**Permissions**  
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional	Choose the IAM role for CloudFormation to use for all operations performed on the stack.	<a href="#">IAM role name</a>	<a href="#">Sample-role-name</a>	Remove
---------------------	--	-------------------------------	----------------------------------	--------

**Advanced options**  
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

- ▶ Stack policy**  
Defines the resources that you want to protect from unintentional updates during a stack update.
- ▶ Rollback configuration**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)
- ▶ Notification options**
- ▶ Stack creation options**

**Feedback** **English (US)**

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

## Default settings

CloudFormation > Stacks > Create stack

**Review eks-vpc**

Step 1: Specify template

Step 2: Specify stack details

Step 3: Configure stack options

Step 4: Review

**Template**

Template URL: <https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon-eks-vpc-sample.yaml>

Stack description: Amazon EKS Sample VPC

Estimate cost: [View](#)

**Step 2: Specify stack details**

**Parameters (4)**

Key	Value
Subnet01Block	192.168.64.0/18
Subnet02Block	192.168.128.0/18
Subnet03Block	192.168.192.0/18
VpcBlock	192.168.0.0/16

**Step 3: Configure stack options**

**Tags (0)**

Key	Value
No tags	
There are no tags defined for this stack	

These are default settings from the URL used. Note the correct 3 subnets in parameters

Feedback English (US)

© 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

**Tags (0)**

Key	Value
No tags	
There are no tags defined for this stack	

**Permissions**

No permissions

There is no IAM role associated with this stack

**Stack policy**

No stack policy

There is no stack policy defined

**Rollback configuration**

Monitoring time: -

CloudWatch alarm ARN: -

**Notification options**

No notification options

These are default settings from the URL used

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The steps are:

- No stack policy**: There is no stack policy defined.
- Rollback configuration**: Monitoring time is set to 1 hour, and CloudWatch alarm ARN is empty.
- Notification options**: No notification options are defined.
- Stack creation options**: Rollback on failure is set to Enabled, Timeout is 1 hour, and Termination protection is set to Disabled.

At the bottom, there is a 'Quick-create link' button, 'Cancel' and 'Previous' buttons, a 'Create change set' button, and a prominent orange 'Create stack' button.

These are default settings from the URL used

The screenshot shows the 'Events' page for the 'eks-vpc' stack in the AWS CloudFormation console. The events table lists 38 entries:

Timestamp	Logical ID	Status	Status reason
2020-03-14 11:39:29 UTC-0700	eks-vpc	CREATE_COMPLETE	-
2020-03-14 11:39:28 UTC-0700	Subnet01RouteTableAssociation	CREATE_COMPLETE	-
2020-03-14 11:39:28 UTC-0700	Subnet02RouteTableAssociation	CREATE_COMPLETE	-
2020-03-14 11:39:27 UTC-0700	Subnet03RouteTableAssociation	CREATE_COMPLETE	-
2020-03-14 11:39:26 UTC-0700	Route	CREATE_COMPLETE	-
2020-03-14 11:39:13 UTC-0700	Subnet01RouteTableAssociation	CREATE_IN_PROGRESS	Resource creation initiated
2020-03-14 11:39:12 UTC-0700	Subnet02RouteTableAssociation	CREATE_IN_PROGRESS	Resource creation initiated
2020-03-14 11:39:12 UTC-0700	Subnet03RouteTableAssociation	CREATE_IN_PROGRESS	Resource creation initiated
2020-03-14 11:39:12 UTC-0700	Subnet01RouteTableAssociation	CREATE_IN_PROGRESS	-
2020-03-14 11:39:12 UTC-0700	Subnet02RouteTableAssociation	CREATE_IN_PROGRESS	-
2020-03-14 11:39:11 UTC-0700	Subnet03RouteTableAssociation	CREATE_IN_PROGRESS	-
2020-03-14 11:39:11 UTC-0700	Route	CREATE_IN_PROGRESS	Resource creation initiated
2020-03-14 11:39:10 UTC-0700	Subnet01	CREATE_COMPLETE	-
2020-03-14 11:39:10 UTC-0700	Subnet02	CREATE_COMPLETE	-
2020-03-14 11:39:10 UTC-0700	Subnet03	CREATE_COMPLETE	-
2020-03-14 11:39:09 UTC-0700	VPCGatewayAttachment	CREATE_COMPLETE	-

At the bottom, there is a 'Feedback' and 'English (US)' button, a copyright notice, and 'Privacy Policy' and 'Terms of Use' links.

Public Stack has been created.

CloudFormation

Stacks

**Stack details**

Drifts

StackSets

Exports

Designer

CloudFormation registry

Resource types

Previous console

Feedback

CloudFormation > Stacks > eks-vpc

eks-vpc

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets

Resources (12)

Logical ID	Physical ID	Type	Status	Status reason
ControlPlaneSecurityGroup	sg-0930183500255351	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
InternetGateway	igw-0e68059751fd1195	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
Route	eks-v-Route-EV2G7X2F13	AWS::EC2::Route	CREATE_COMPLETE	-
RouteTable	rtb-08a015ad6662b5f23	AWS::EC2::RouteTable	CREATE_COMPLETE	-
Subnet01	subnet-0ff4bb49f9050a79	AWS::EC2::Subnet	CREATE_COMPLETE	-
Subnet01RouteTableAssociation	rtbassoc-0c223399dd8de0ad9	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
Subnet02	subnet-06544cc4d42ab0b21	AWS::EC2::Subnet	CREATE_COMPLETE	-
Subnet02RouteTableAssociation	rtbassoc-07af9bd1710a7b2f7	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
Subnet03	subnet-093dfb2ef847731a2	AWS::EC2::Subnet	CREATE_COMPLETE	-
Subnet03RouteTableAssociation	rtbassoc-0867547c8c0c3ab56	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
VPC	vpc-0ca954e021ca1f0a	AWS::EC2::VPC	CREATE_COMPLETE	-
VPCGatewayAttachment	eks-v-VPCG-1C37EM2LAC20K	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-

https://us-west-2.console.aws.amazon.com/cloudformation/home?region=us-west-2#

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Resources for stack. Note the 3 subnets

Amazon Container Services

Amazon ECS

Clusters

Task definitions

**Amazon EKS**

Clusters

Amazon ECR

Repositories

Compute

## Elastic Kubernetes Service (Amazon EKS)

Fully managed Kubernetes control plane

Amazon EKS is a managed service that makes it easy for you to use Kubernetes on AWS without needing to install and operate your own Kubernetes control plane.

How it works

Why use Amazon EKS?

Amazon EKS exposes a Kubernetes API endpoint. Your existing Kubernetes tooling can connect directly to EKS managed control plane. Worker nodes run as EC2 instances in your account.

Create EKS cluster

Cluster name: eks-cluster

Next step

Pricing (US)

EKS Control Plane: \$0.10 USD (per hour)

Worker nodes: EC2 Pricing

Getting started

For more details, see the Amazon EKS product page.

For a walkthrough of deploying an EKS cluster, see Getting Started.

Amazon EKS resources

Documentation

API Reference

Amazon EKS FAQ

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

eks-cluster

**General configuration**

Cluster name: eks-cluster

Kubernetes version: 1.15

Role name: eks-role

**Networking**

VPC: vpc-0ca954e021c8a10a - 192.168.0.0/16

Subnets:

Subnet	Name	Availability Zone	Subnet IPv4 CIDR
subnet-06544ccde2ab0b21	eks-vpc-Subnet02	us-west-2b	192.168.128.0/18
subnet-0ff4bb4990580a79	eks-vpc-Subnet01	us-west-2a	192.168.64.0/18
subnet-093dfb2ef847731a2	eks-vpc-Subnet03	us-west-2c	192.168.192.0/18

Security groups:

Group	Name	Description
sg-07ae0de18967ebff	default	default VPC security group
sg-0c930183500235351	eks-vpc-ControlPlaneSecurityGroup-LPGGGV932DA482	Cluster communication with worker nodes

**Feedback** English (US)

Choosing the stack just created eks-vpcpub-stack along with Kubernetes 1.15 and the created service role

**API server endpoint access**

Configure access to the Kubernetes API server endpoint.

Private access: Configure access to the API server endpoint from within your VPC. (Disabled)

Public access: Configure access to the API server endpoint from outside of your VPC. (Enabled)

**Secrets encryption**

Enable envelope encryption of Kubernetes secrets using KMS.

**Logging**

CloudWatch log group: EKS automatically creates a CloudWatch log group for you when you enable logging.

**Feedback** English (US)

All subnets selected to distribute worker nodes for improved redundancy. Allocated security group from the given template allows the cluster control plane to communicate with worker nodes.

**Secrets encryption** Info

Enable envelope encryption of Kubernetes secrets using KMS  
Enable envelope encryption to provide an additional layer of encryption for your Kubernetes secrets.

**Logging**

CloudWatch log group  
EKS automatically creates a CloudWatch log group for you when you enable logging.

API server  
Logs pertaining to API requests to the cluster.  
 Disabled

Audit  
Logs pertaining to cluster access via the Kubernetes API.  
 Disabled

Authenticator  
Logs pertaining to authentication requests into the cluster.  
 Disabled

Controller manager  
Logs pertaining to state of cluster controllers.  
 Disabled

Scheduler  
Logs pertaining to scheduling decisions.  
 Disabled

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources. Tags are optional.

This cluster does not have any tags.

Add tag

Remaining tags available to add: 50

Cancel **Create**

All other settings left at default

**eks-cluster**

**General configuration**

Kubernetes version: 1.15 Platform version: eks.1 Status: Active

Certificate authority: L50hL51CRUjdJtB0DRVAlJSUzJQOFURSOhL50hOk1JSUN5RENDQWJDZFS5UJBZ01CQURBTkJna5Foa2lHOXowQkFRc0ZBREFWTJNjd0vRwURUvUVUFERxvcmRXSmwvY201bGRHvnpNQRyRFRj0d01ETXhOREUOTIRvE1Wb1HEVE13TURNnE1gRTROVF4tV2vd02URVRNQKvhQTFVRQjbeE1LYTNWaVpYSnVaWFaY3pDQ0FTSxgEUvIKS23aSWh2Y05BUUVCFBRGdnRVBBRENDQ

Cluster ARN: arn:aws:eks:us-west-2:758287676861:cluster/eks-cluster

Cluster IAM Role ARN: arn:aws:iam::758287676861:role/eks-role

**Node Groups (0)** Info

Group name	▲ Desired size	▼ AMI release version	▼ Status
No Managed Node Groups			
Nodes that are not part of an Amazon EKS Managed Node Group are not shown in the AWS console.			
<b>Add Node Group</b>			

**Networking**

VPC: vpc-0ca954e021cda1f0a	Subnets: subnet-06544ccde42ab0b21, subnet-0ff4b0a49f90580a79	Cluster security group: sg-09bd32c2d661ba352	API server endpoint access: Private access, Disabled
----------------------------	--	--	--

Eks-cluster created

```

Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

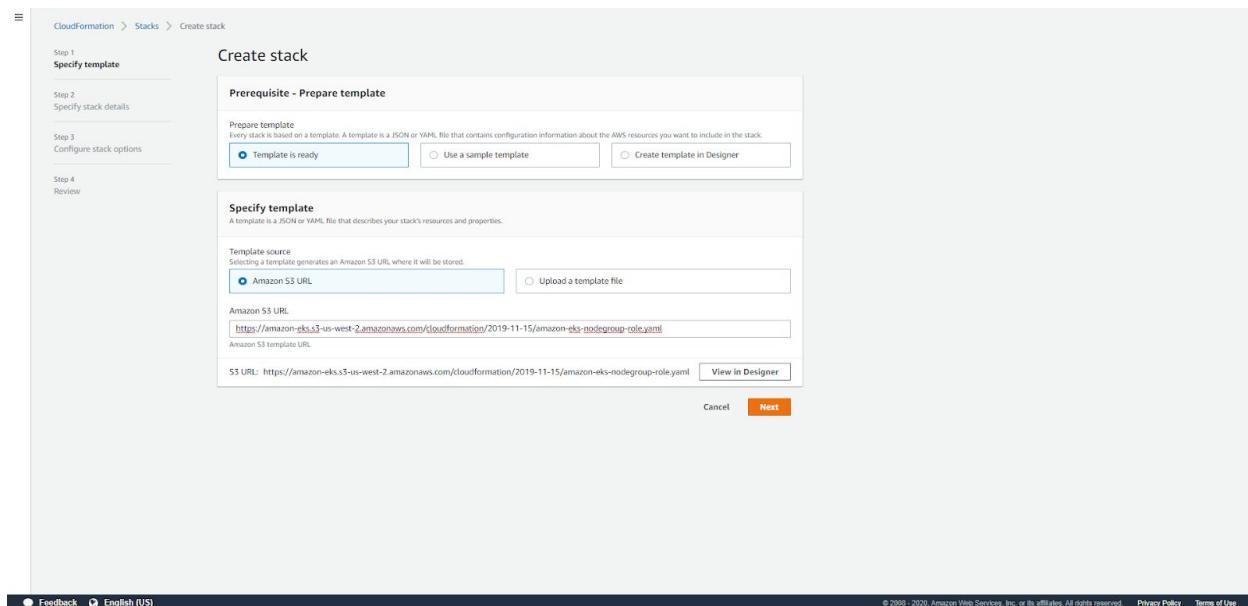
C:\Users\CalBright>aws eks --region us-west-2 update-kubeconfig --name eks-cluster
Updated context arn:aws:eks:us-west-2:758287676861:cluster/eks-cluster in C:\Users\CalBright\.kube\config

C:\Users\CalBright>kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.100.0.1    <none>        443/TCP     113m

C:\Users\CalBright>

```

Aws cli has pointed eks to the active cluster just created



Template file<sup>3</sup> will aid creation of stack with default values to get started

<sup>3</sup> <https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html>

CloudFormation > Stacks > Create stack

Step 1 Specify template

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

### Specify stack details

**Stack name**

Stack name: eks-node-group-instance-role  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters  
There are no parameters defined in your template

Cancel Previous Next

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## Eks-node-group-instance-role

CloudFormation > Stacks > Create stack

Step 1 Specify template

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

### Configure stack options

**Tags**

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove
Add tag		

**Permissions**

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name: ▾	Sample-role-name	Remove
------------------	------------------	--------

**Advanced options**

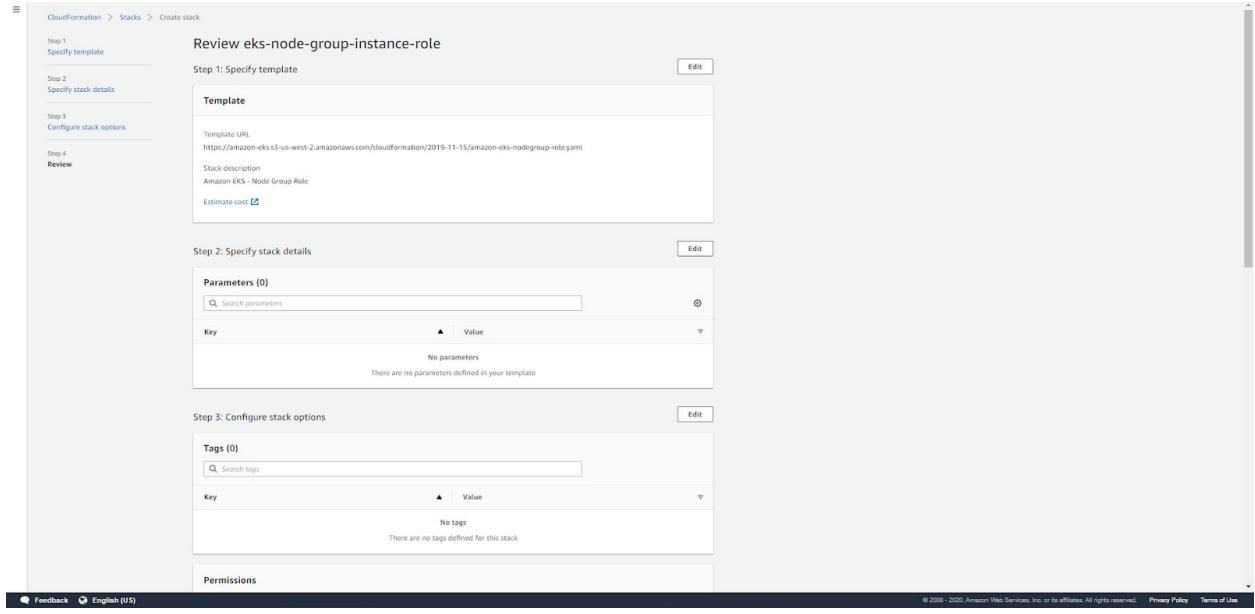
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

- ▶ Stack policy  
Defines the resources that you want to protect from unintentional updates during a stack update.
- ▶ Rollback configuration  
Specifies alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)
- ▶ Notification options
- ▶ Stack creation options

Cancel Previous Next

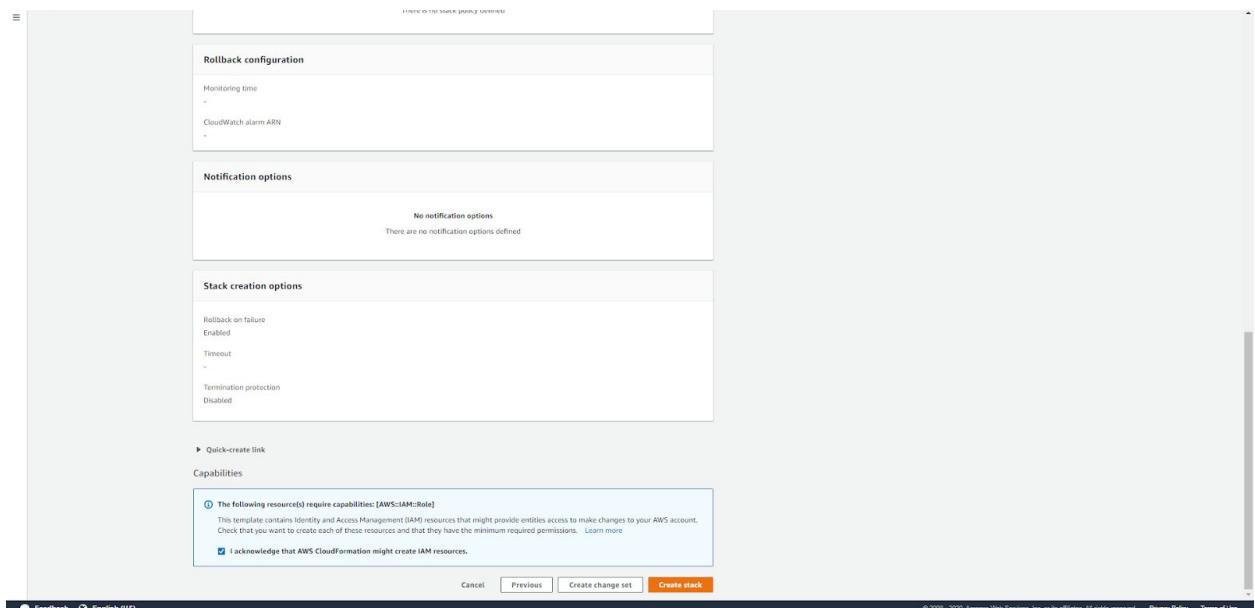
Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Defaults left as is



The screenshot shows the 'Review eks-node-group-instance-role' step of the CloudFormation wizard. It displays the template URL (<https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon-eks-nodegroup-role.yaml>), stack description ('Amazon EKS - Node Group Role'), and an estimate cost link. Below this, the 'Step 2: Specify stack details' section shows parameters and tags, both currently empty. The 'Step 3: Configure stack options' section also shows empty tags and permissions. At the bottom, there are links for Feedback, English (US), and a note about the review process.

## Node group review



This screenshot shows the final review step before creating the node stack. It includes sections for 'Rollback configuration' (monitoring time and CloudWatch alarm ARN), 'Notification options' (no notification options defined), and 'Stack creation options' (rollback on failure, timeout, and termination protection). A note at the bottom states that the template requires IAM capabilities and includes a checkbox for acknowledging this. Navigation buttons at the bottom include 'Cancel', 'Previous', 'Create change set', and 'Create stack'.

## Review before creating node stack

**CloudFormation**

**Stacks (2)**

**Events (5)**

Timestamp	Logical ID	Status	Status reason
2020-03-14 13:58:50 UTC-0700	eks-node-group-instance-role	CREATE_COMPLETE	-
2020-03-14 13:58:48 UTC-0700	NodeInstanceRole	CREATE_COMPLETE	-
2020-03-14 13:58:34 UTC-0700	NodeInstanceRole	CREATE_IN_PROGRESS	Resource creation initiated
2020-03-14 13:58:33 UTC-0700	NodeInstanceRole	CREATE_IN_PROGRESS	-
2020-03-14 13:58:30 UTC-0700	eks-node-group-instance-role	CREATE_IN_PROGRESS	User initiated

EKS node group has been created

**EKS**

**Clusters**

**eks-cluster**

**General configuration**

Kubernetes version: 1.15 Platform version: eks-1 Status: Active

API server endpoint: https://0A76A9198A0C2011D9360F225814476.gr7.us-west-2.eks.amazonaws.com

OpenID Connect provider URL: https://oidc.eks.us-west-2.amazonaws.com/id/0A76A9198A0C2011D9360F225814476

Cluster ARN: arn:aws:eks:us-west-2:758287676861:cluster/eks-cluster

Cluster IAM Role ARN: arn:aws:iam::758287676861:role/eks-role

**Node Groups (0) Info**

**Networking**

VPC: vpc-0ca954e021d8a1f0s

Subnets:

- subnet-06544cc9462ab0b21
- subnet-0ff4bb-0950580a79
- subnet-0950ff2f847731a2

Cluster security group info: sg-0f0eb12a2de61ba152

Additional security groups: sg-0-930183500235351

API server endpoint access:

- Private access: Disabled
- Public access: Enabled
- Public access source whitelist: 0.0.0.0/0 (open to all traffic)

Adding a Node Group in eks-cluster

**Configure Node Group**

A Node Group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple Node Groups to your cluster. [Info](#)

**Group configuration**  
These properties cannot be changed after the Node Group is created.

**Name**  
Assign a unique name for this Node Group.

**Node IAM Role Name** [Info](#)  
Select the IAM Role that will be used by the nodes.

**Subnets** [Info](#)  
Specify the subnets in your VPC where your nodes will run.

**Allow remote access to nodes** [Info](#)  
Without remote access enabled you will not be able to directly connect to nodes after they are created.

**SSH key pair** [Info](#)  
Select an SSH key pair to allow secure remote access to your nodes.

**Allow remote access from**  
Configure the source IP ranges that can remotely access nodes.

All  
Do not restrict source IPs that can remotely access nodes.

Selected security groups  
Specify security groups to restrict which source IPs can remotely access nodes.

**Tags and labels**

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Note that the new keypair made for this Node Group

EKS > Clusters > eks-cluster > Add Node Group

**Step 1** Configure Node Group

**Step 2** Set compute configuration

**Step 3** Set scaling configuration

**Step 4** Review and create

**Set compute configuration**

**Node compute configuration**  
These properties cannot be changed after the Node Group is created.

**AMI type** [Info](#)  
Select the EKS-optimized Amazon Machine Image for nodes.

**Instance type** [Info](#)  
Select the EC2 instance type for nodes.

**Disk size**  
Select the size of the attached EBS volume for each node.  
 GiB

Cancel

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

T3micro and default settings

**Amazon Container Services**

EKS > Clusters > eks-cluster > Add Node Group

Step 1 Configure Node Group

Step 2 Set compute configuration

Step 3 Set scaling configuration

Step 4 Review and create

**Set scaling configuration**

**Group size**

Minimum size  
Set the minimum number of nodes that the group can scale in to.  
 nodes

Maximum size  
Set the maximum number of nodes that the group can scale out to.  
 nodes

Desired size  
Set the desired number of nodes that the group should launch with initially.  
 nodes

Cancel Previous **Next**

Desired load balancing size set to 3

**Amazon Container Services**

Step 3 Set scaling configuration

Step 4 Review and create

Name: eks-cluster-nodes

Node IAM Role Name: eks-node-group-instance-role-NodeInstanceRole-13TEG70F4W0CH

Subnets:  
subnet-06544c04e2a0b0b21  
subnet-0ff4ab49f90580b79  
subnet-095affb2af547731a2

Allow remote access to nodes: Enabled

SSH key pair: eks-node-group-keypair

Allow remote access from: All

**Tags and labels**

- Tags (0)
- Kubernetes labels (0)

**Step 2: Set compute configuration**

**Node compute configuration**

AMI type: Amazon Linux 2 (AL2\_x86\_64)

Instance type: t3.micro

Disk size: 20

**Step 3: Set scaling configuration**

**Group size**

Minimum size	Maximum size	Desired size
2 nodes	5 nodes	5 nodes

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Review page before launch

**Amazon Container Services**

EKS > Clusters > eks-cluster

### eks-cluster

**General configuration**

Kubernetes version 1.15	Platform version eks.1	Status <span style="color: green;">Active</span>
API server endpoint <a href="https://0A768AF9A30C2D11FD930F225814476.gr7.us-west-2.eks.amazonaws.com">https://0A768AF9A30C2D11FD930F225814476.gr7.us-west-2.eks.amazonaws.com</a>	Certificate authority <a href="#">View details</a>	
OpenID Connect provider URL <a href="https://idc.eks.us-west-2.amazonaws.com/id/0A768AF9A30C2D11FD930F225814476">https://idc.eks.us-west-2.amazonaws.com/id/0A768AF9A30C2D11FD930F225814476</a>		
Cluster ARN <a href="#">arn:aws:eks:us-west-2:758287676861:cluster/eks-cluster</a>	Cluster IAM Role ARN <a href="#">arn:aws:iam::758287676861:role/eks-role</a>	

**Node Groups (1) [Info](#)**

Group name	Desired size	AMI release version	Status
eks-cluster-nodes	3	1.15.10-20200228	<span style="color: green;">Active</span>

**Networking**

VPC <a href="#">vpc-0ca954e021d1f0a</a>	Subnets <a href="#">subnet-06544cd4e2ab0b21</a> <a href="#">subnet-0f4bb49f90580a79</a> <a href="#">subnet-095dfb2ef847731a2</a>	Cluster security group <a href="#">Info</a> sg-05db32c2de61ba352 Additional security groups sg-0c930183500235351	API server endpoint access Private access Disabled Public access Enabled Public access source whitelist
--	---	--	--

**Feedback** [English \(US\)](#)

© 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Eks-cluster-nodes is now active

**Amazon Container Services**

EKS > Clusters > eks-cluster > Node Group : eks-cluster-nodes

### eks-cluster-nodes

**Node Group configuration**

Kubernetes version 1.15	AMI type <a href="#">Info</a> AL2_x64_64	Status <span style="color: green;">Active</span>
Instance type t3.micro	AMI release version <a href="#">Info</a> 1.15.10-20200228	Disk size 20 GiB

**Details** [Health issues](#) [Kubernetes labels](#) [Updates](#) [Tags](#)

**Details**

Node Group ARN <a href="#">arn:aws:eks:us-west-2:758287676861:nodegroup/eks-cluster/eks-cluster-nodes/e6b86d73-707a-55fb-18e4-3c329076a503</a>	Autoscaling group name <a href="#">eks-6bb86d73-707a-55fb-18e4-3c329076a503</a>	Minimum size 2 nodes	Subnets <a href="#">subnet-06544cd4e2ab0b21</a> <a href="#">subnet-0f4bb49f90580a79</a> <a href="#">subnet-095dfb2ef847731a2</a>
Launch time Mar 14th 2020 at 2:15 PM	Node IAM Role Name <a href="#">eks-node-group-instance-role-NodeInstanceRole-13TE070FWOCH</a>	Maximum size 5 nodes	Allow remote access to nodes Enabled
		Desired size 3 nodes	SSH key pair <a href="#">eks-node-group-keypair</a>
			Allow remote access from All

**Feedback** [English \(US\)](#)

© 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

There are currently zero health issues for the nodes.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area displays a table of instances. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, IPv6 IPs, Key Name, Monitoring, Launch Time, and Security. There are three entries in the table:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time	Security
i-0313b789d223aee	t3.micro	us-west-2b	running	2/2 checks ...	None		ec2-34-214-104-230.us.	34.214.104.230	-	eks-node-grou...	disabled	March 14, 2020 at 2:16 59 P...	eks-remo...
i-051fb10fcf2d3206	t3.micro	us-west-2c	running	2/2 checks ...	None		ec2-34-212-139-160.us.	34.212.139.160	-	eks-node-grou...	disabled	March 14, 2020 at 2:16 59 P...	eks-remo...
i-053f6395a0fd79f76	t3.micro	us-west-2a	running	2/2 checks ...	None		ec2-34-223-229-151.us...	34.223.229.151	-	eks-node-grou...	disabled	March 14, 2020 at 2:16 59 P...	eks-remo...

## Deployed worker nodes in EC2

```

Command Prompt - kubectl get nodes --watch
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

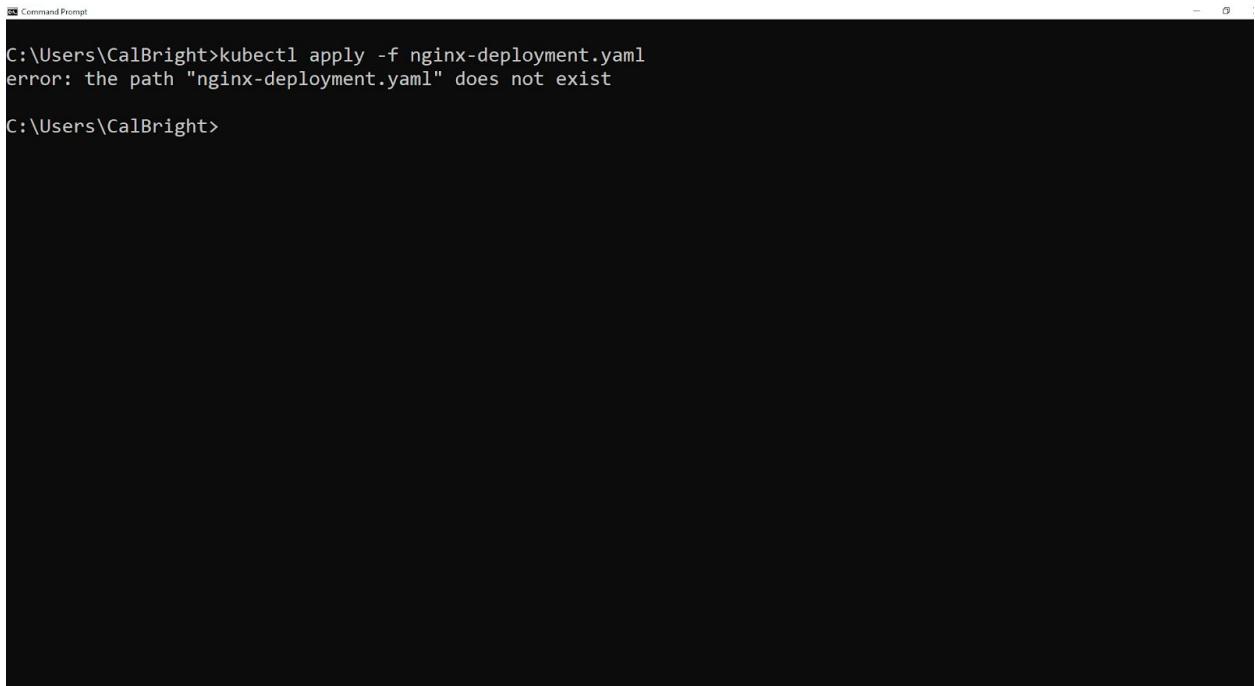
C:\Users\CalBright>aws eks --region us-west-2 update-kubeconfig --name eks-cluster
Updated context arn:aws:eks:us-west-2:758287676861:cluster/eks-cluster in C:\Users\CalBright\.kube\config

C:\Users\CalBright>kubectl get svc
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP  10.100.0.1  <none>        443/TCP   113m

C:\Users\CalBright>kubectl get nodes --watch
NAME                  STATUS  ROLES   AGE     VERSION
ip-192-168-101-195.us-west-2.compute.internal  Ready   <none>  6m52s  v1.15.10-eks-bac369
ip-192-168-191-13.us-west-2.compute.internal    Ready   <none>  6m38s  v1.15.10-eks-bac369
ip-192-168-192-38.us-west-2.compute.internal    Ready   <none>  6m47s  v1.15.10-eks-bac369

```

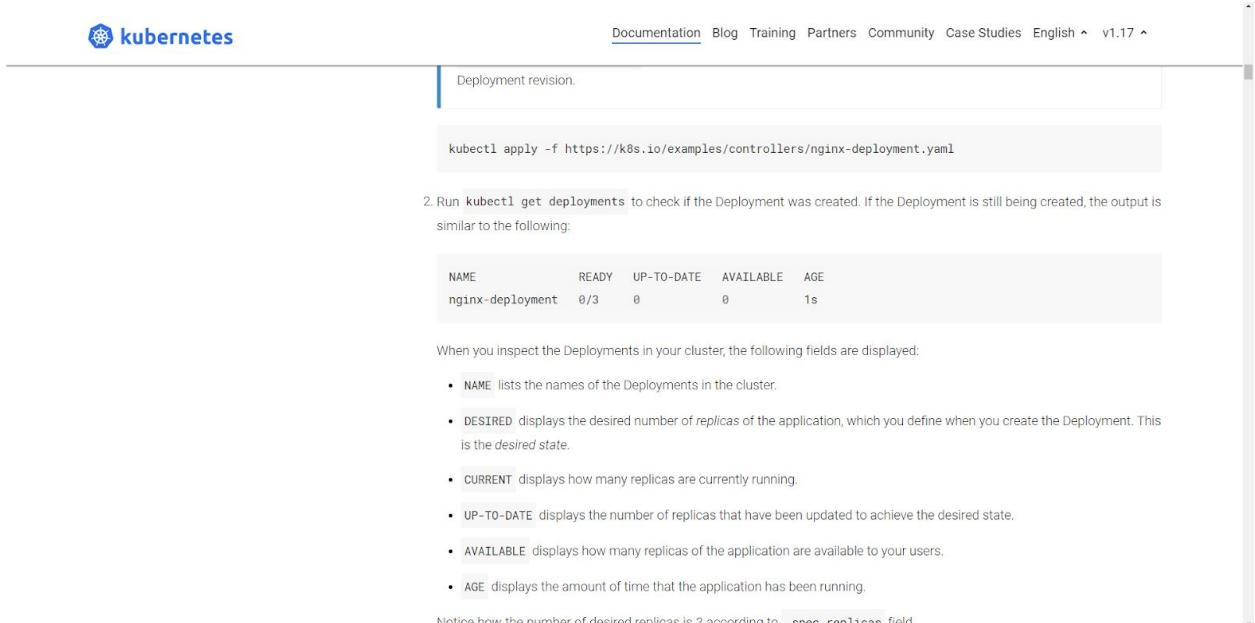
3 worker nodes ready for container applications



```
C:\Users\CalBright>kubectl apply -f nginx-deployment.yaml
error: the path "nginx-deployment.yaml" does not exist

C:\Users\CalBright>
```

However the yaml file does not exist



The screenshot shows a section of the Kubernetes documentation titled "Deployment revision." It includes a code example for applying a deployment configuration from a URL:

```
kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml
```

Below the code example, there is a note: "2. Run `kubectl get deployments` to check if the Deployment was created. If the Deployment is still being created, the output is similar to the following."

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	0/3	0	0	1s

Following this, there is a list of deployment fields:

- NAME: lists the names of the Deployments in the cluster.
- DESIRED: displays the desired number of *replicas* of the application, which you define when you create the Deployment. This is the *desired state*.
- CURRENT: displays how many replicas are currently running.
- UP-TO-DATE: displays the number of replicas that have been updated to achieve the desired state.
- AVAILABLE: displays how many replicas of the application are available to your users.
- AGE: displays the amount of time that the application has been running.

Notice how the number of desired replicas is 3 according to `spec.replicas` field

Searching AWS getting started site I find that Kubernetes<sup>4</sup> has a nginx Deployment guide

<sup>4</sup> <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/#creating-a-deployment>

```
C:\Users\CalBright>kubectl apply -f nginx-deployment.yaml
error: the path "nginx-deployment.yaml" does not exist

C:\Users\CalBright>kubectl describe deployments

C:\Users\CalBright>kubectl get pods
No resources found in default namespace.

C:\Users\CalBright>kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml
deployment.apps/nginx-deployment created

C:\Users\CalBright>
```

Success!

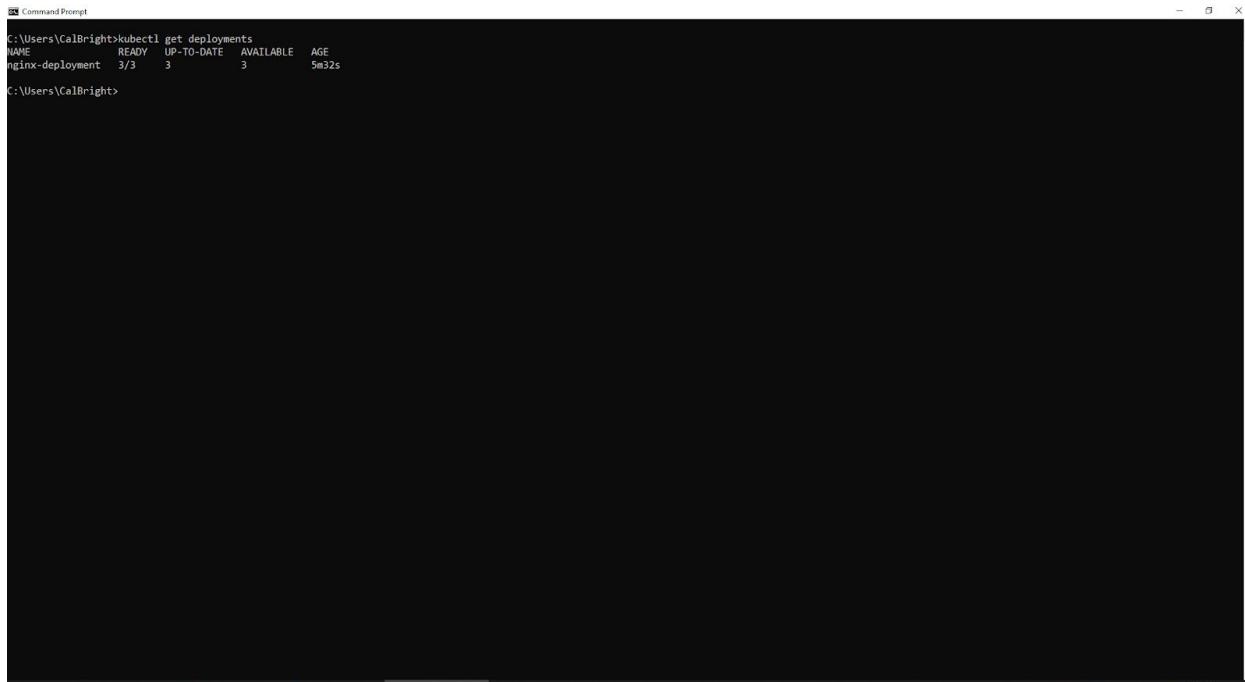
```
C:\Users\CalBright>kubectl get all -o wide
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE   IP           NODE      NOMINATED-NODE  READINESS   GATES
pod/nginx-deployment-5754944d6c-2fw99  1/1    Running     0          80s    192.168.156.136 ip-192-168-191-13.us-west-2.compute.internal  <none>       <none>
pod/nginx-deployment-5754944d6c-5wfg4  1/1    Running     0          80s    192.168.81.67  ip-192-168-101-195.us-west-2.compute.internal  <none>       <none>
pod/nginx-deployment-5754944d6c-kbb5x  1/1    Running     0          80s    192.168.86.146 ip-192-168-101-195.us-west-2.compute.internal  <none>       <none>

NAME            READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES   SELECTOR
deployment.apps/nginx-deployment  3/3     3           3          80s   nginx        nginx:1.7.9  app=nginx
NAME            DESIRED  CURRENT   READY   AGE   CONTAINERS   IMAGES   SELECTOR
replicaset.apps/nginx-deployment-5754944d6c  3       3           3          80s   nginx        nginx:1.7.9  app=nginx,pod-template-hash=5754944d6c

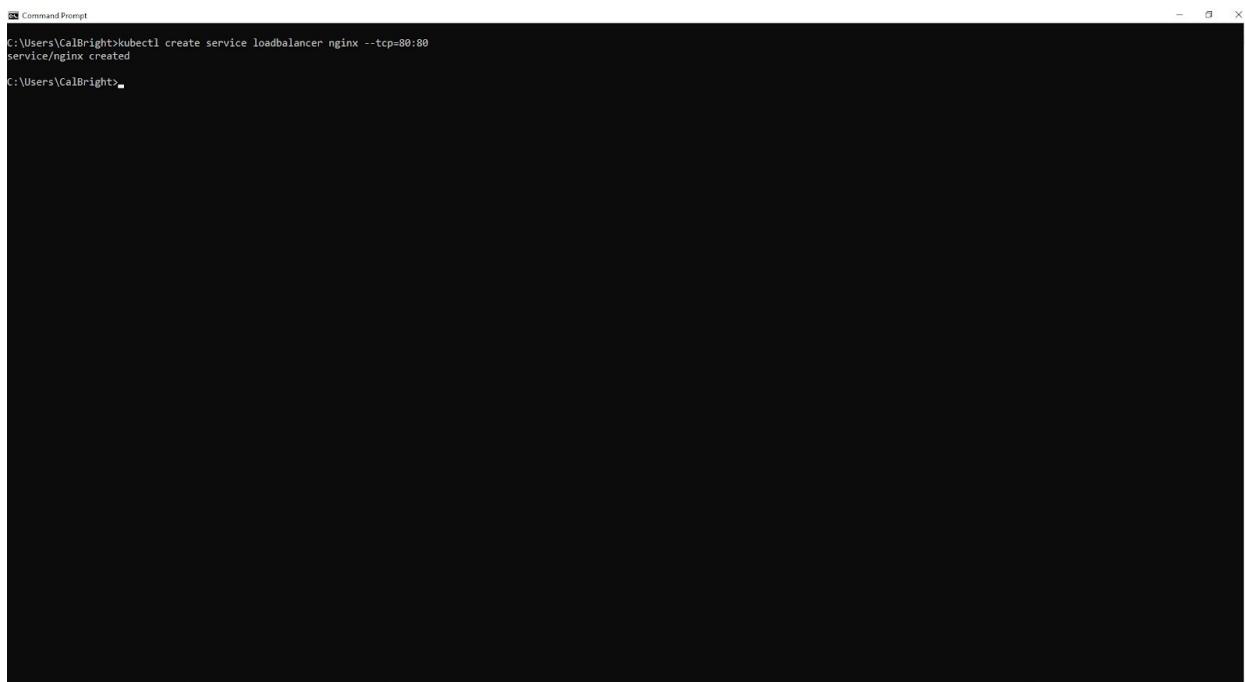
C:\Users\CalBright>
```

Worker nodes are running 3 nginx running



```
C:\Users\CalBright>kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  3/3     3          3           5m32s
C:\Users\CalBright>
```

Deployments have been created



```
C:\Users\CalBright>kubectl create service loadbalancer nginx --tcp=80:80
service/nginx created
C:\Users\CalBright>
```

A load balancing service is created

```

Select Command Prompt
C:\Users\CalBright>kubectl create service loadbalancer nginx --tcp=80:80
service/nginx created

C:\Users\CalBright>kubectl get all -o wide
NAME                                     READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
pod/nginx-deployment-5754944d6c-2fw9q   1/1    Running   0          20m   192.168.156.136   ip-192-168-191-13.us-west-2.compute.internal   <none>        <none>
pod/nginx-deployment-5754944d6c-5wfq4   1/1    Running   0          20m   192.168.81.67    ip-192-168-101-195.us-west-2.compute.internal   <none>        <none>
pod/nginx-deployment-5754944d6c-kbb5x   1/1    Running   0          20m   192.168.86.146   ip-192-168-101-195.us-west-2.compute.internal   <none>        <none>

NAME          TYPE    CLUSTER-IP   EXTERNAL-IP
service/kubernetes   ClusterIP   10.100.0.1   <none>
service/nginx     LoadBalancer  10.100.11.24  a82a01bc51983456ab75231870960aeef-1888002545.us-west-2.elb.amazonaws.com

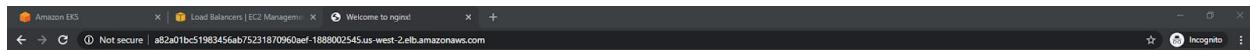
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES          SELECTOR
deployment.apps/nginx-deployment-5754944d6c  3/3     3           3          20m   nginx        nginx:1.7.9   app=nginx
replicaset.apps/nginx-deployment-5754944d6c  3       3           3          20m   nginx        nginx:1.7.9   app=nginx,pod-template-hash=5754944d6c

C:\Users\CalBright>

```

LoadBalancer is running

Elastic load balanced created via EKS cluster using attached role



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

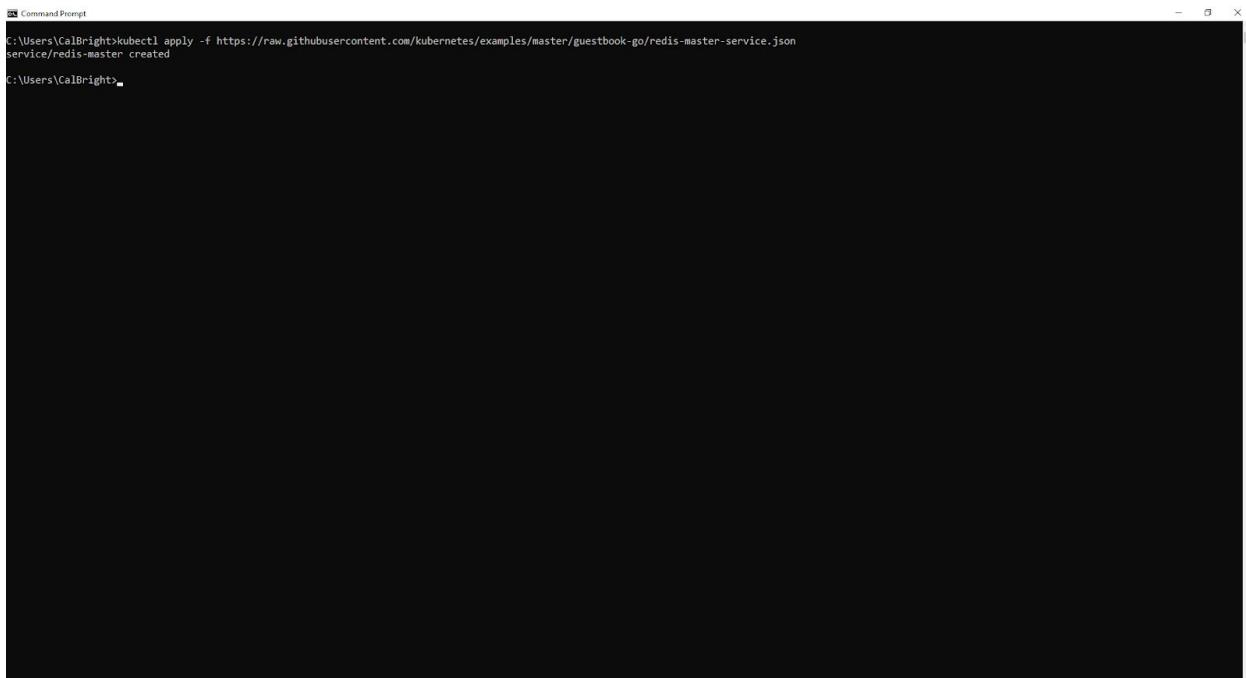
For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

nginx welcome page

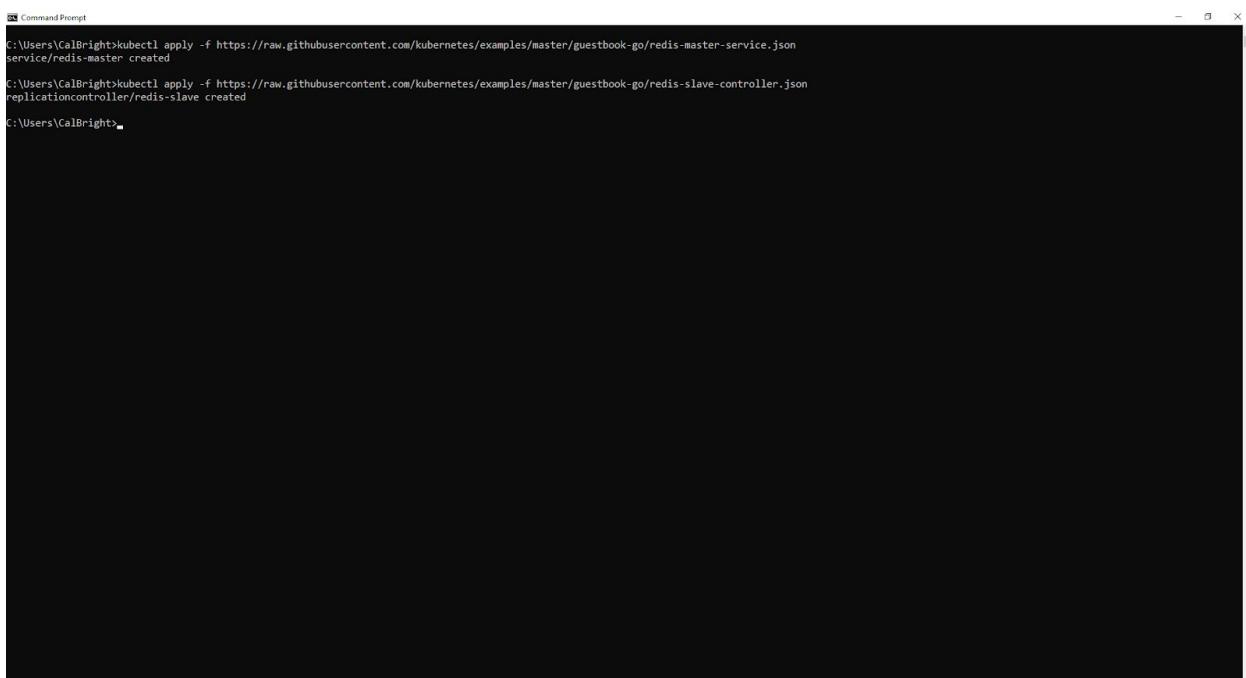
```
PS C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-controller.json
replicationcontroller/redis-master created
C:\Users\CalBright>
```

Next, moving on to the guestbook app. Here the replication controller is created



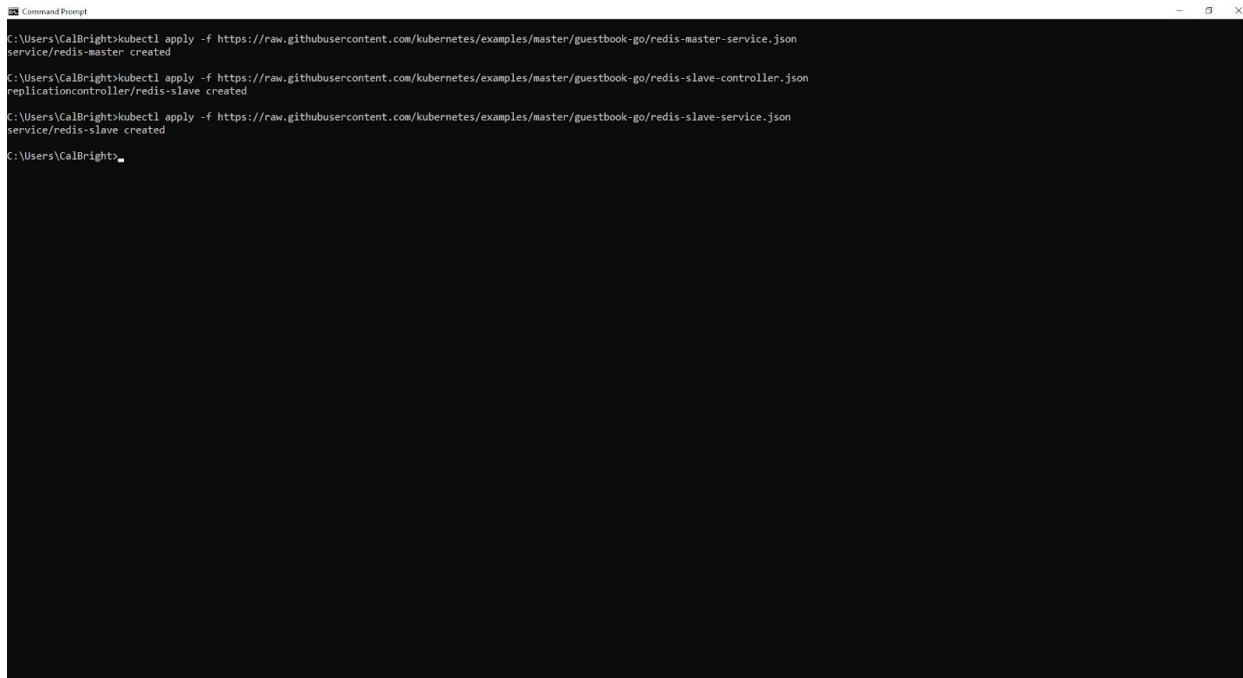
```
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json
service/redis-master created
C:\Users\CalBright>
```

Master service for guestbook app



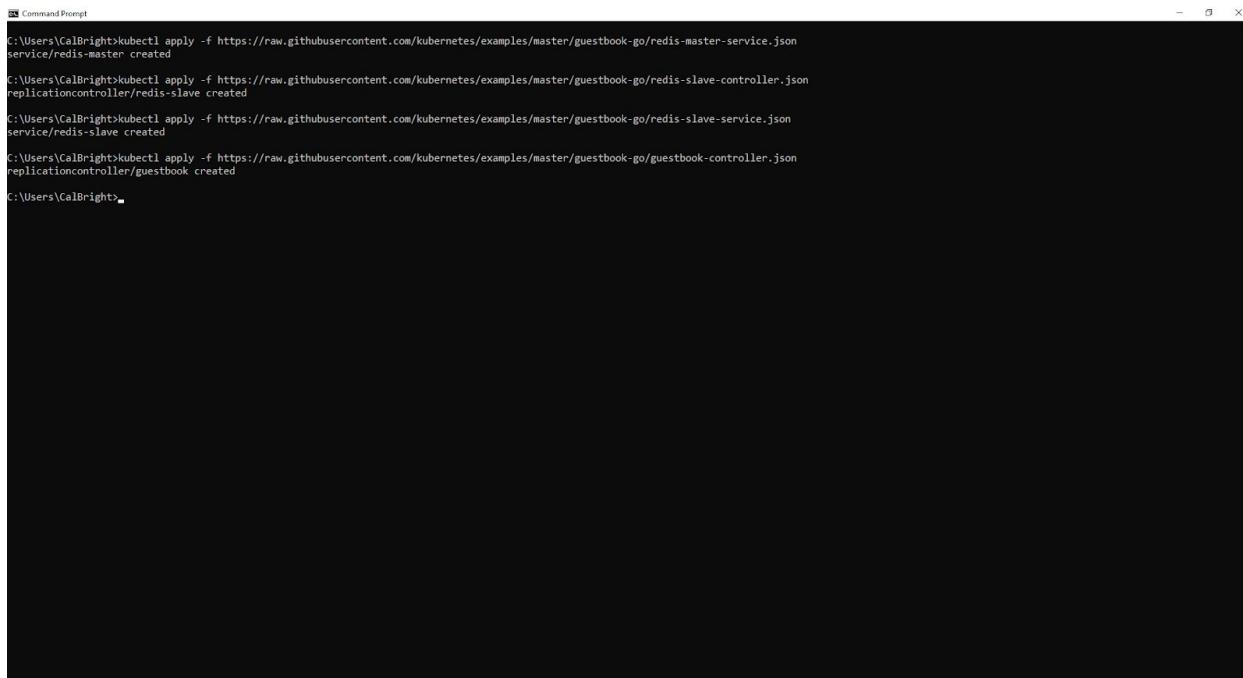
```
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json
service/redis-master created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json
replicationcontroller/redis-slave created
C:\Users\CalBright>
```

Replication slave for guestbook app



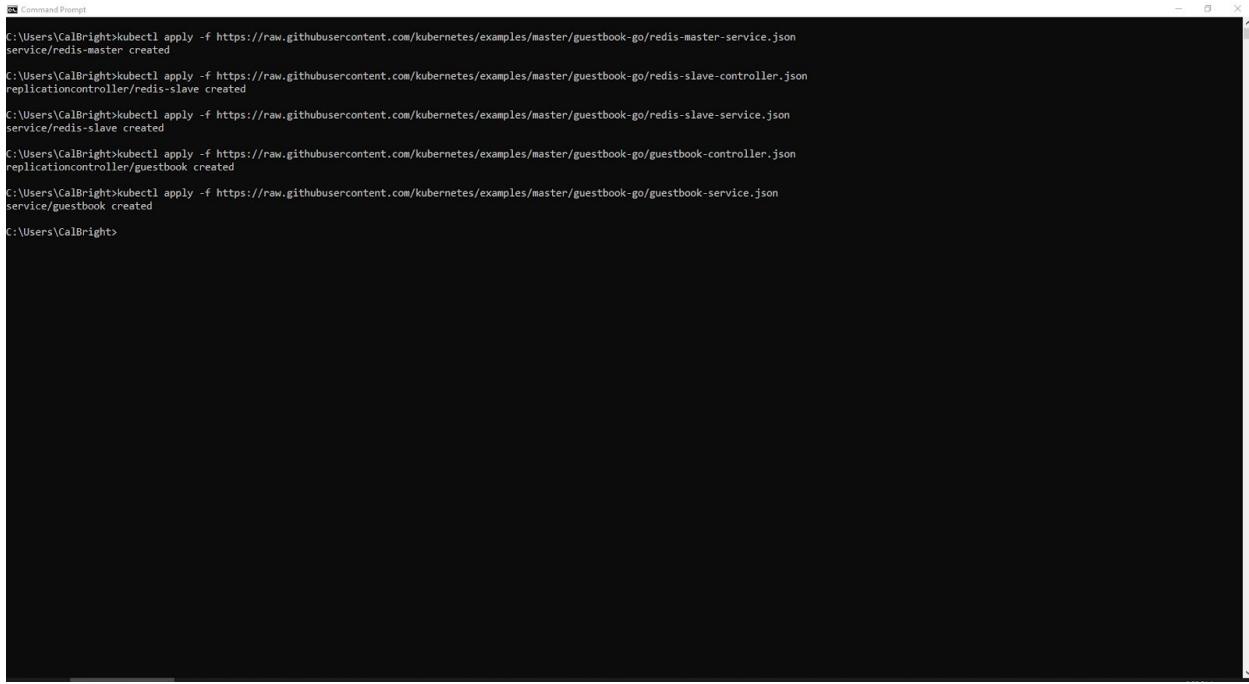
```
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json
service/redis-master created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json
replicationcontroller/redis-slave created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-service.json
service/redis-slave created
C:\Users\CalBright>
```

slave service is now created



```
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json
service/redis-master created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json
replicationcontroller/redis-slave created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-service.json
service/redis-slave created
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-controller.json
replicationcontroller/guestbook created
C:\Users\CalBright>
```

Guestbook replication controller created



```
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json
service/redis-master created

C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json
replicationcontroller/redis-slave created

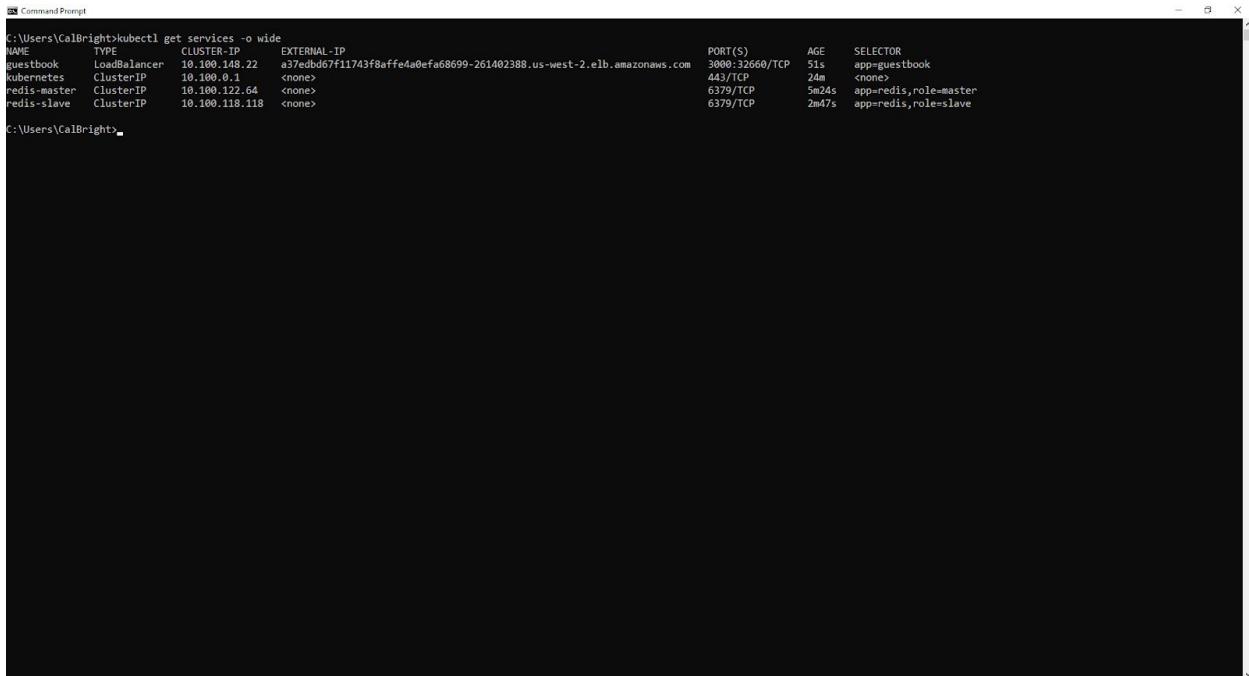
C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-service.json
service/redis-slave created

C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-controller.json
replicationcontroller/guestbook created

C:\Users\CalBright>kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-service.json
service/guestbook created

C:\Users\CalBright>
```

Finally the guestbook service itself is created



```
C:\Users\CalBright>kubectl get services --wide
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE      SELECTOR
guestbook  LoadBalancer  10.100.148.22  a37dbd67f11743f8affe4a0efa68699-261402388.us-west-2.elb.amazonaws.com  3000:32660/TCP  51s      app=guestbook
kubernetes  ClusterIP   10.100.0.1    <none>        443/TCP     24m      <none>
redis-master  ClusterIP   10.100.122.64  <none>        6379/TCP    5m24s    app=redis,role=master
redis-slave   ClusterIP   10.100.118.118  <none>        6379/TCP    2m47s    app=redis,role=slave

C:\Users\CalBright>
```

The guestbook app has been provisioned

# Guestbook

Hello world

Hello world

SUBMIT

http://a37edb67f11743f8affe4a0efa68699-261402388.us-west-2.elb.amazonaws.com:3000/  
[/env](#) [/info](#)

The guestbook is up and running

```
C:\Users\CalBright>kubectl get services -o wide
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE   SELECTOR
guestbook      LoadBalancer 10.100.148.22  a37edb67f11743f8affe4a0efa68699-261402388.us-west-2.elb.amazonaws.com  3000:32660/TCP  51s  app=guestbook
kubernetes     ClusterIP   10.100.0.1    <none>          443/TCP       24m  <none>
redis-master   ClusterIP   10.100.122.64  <none>          6379/TCP     5m24s app=redis,role=master
redis-slave    ClusterIP   10.100.118.118  <none>          6379/TCP     2m47s app=redis,role=slave

C:\Users\CalBright>kubectl get services -o wide
NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE   SELECTOR
guestbook      LoadBalancer 10.100.148.22  a37edb67f11743f8affe4a0efa68699-261402388.us-west-2.elb.amazonaws.com  3000:32660/TCP  3m46s app=guestbook
kubernetes     ClusterIP   10.100.0.1    <none>          443/TCP       27m  <none>
redis-master   ClusterIP   10.100.122.64  <none>          6379/TCP     8m19s app=redis,role=master
redis-slave    ClusterIP   10.100.118.118  <none>          6379/TCP     5m42s app=redis,role=slave

C:\Users\CalBright>kubectl delete rc/redis-master rc/redis-slave rc/guestbook svc/redis-master svc/redis-slave svc/guestbook
replicationcontroller "redis-master" deleted
replicationcontroller "redis-slave" deleted
replicationcontroller "guestbook" deleted
service "redis-master" deleted
service "redis-slave" deleted
service "guestbook" deleted

C:\Users\CalBright>
```

Deletion of all guestbook replicators and services