

AWS – Elastic Kubernetes Service (Amazon EKS) – Hands-on guide/ POC/ Case Study/ Use Case.

Mr. Subramanyam Tirumani Vemala

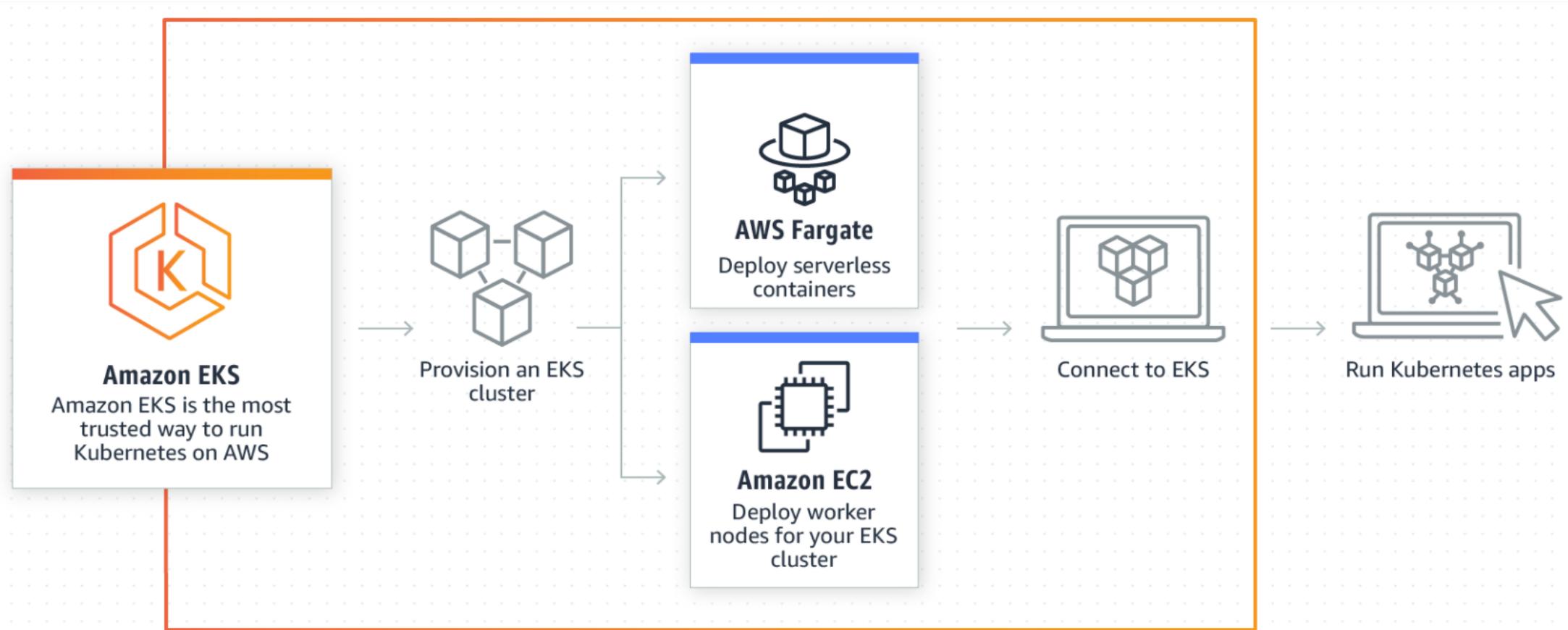
<https://www.linkedin.com/in/subbulinkedin/>



EKS Architecture - Component diagram: In this POC, TypeB (Amazon EC2's) is used, as we need to go for Cloud agnostic Architecture.

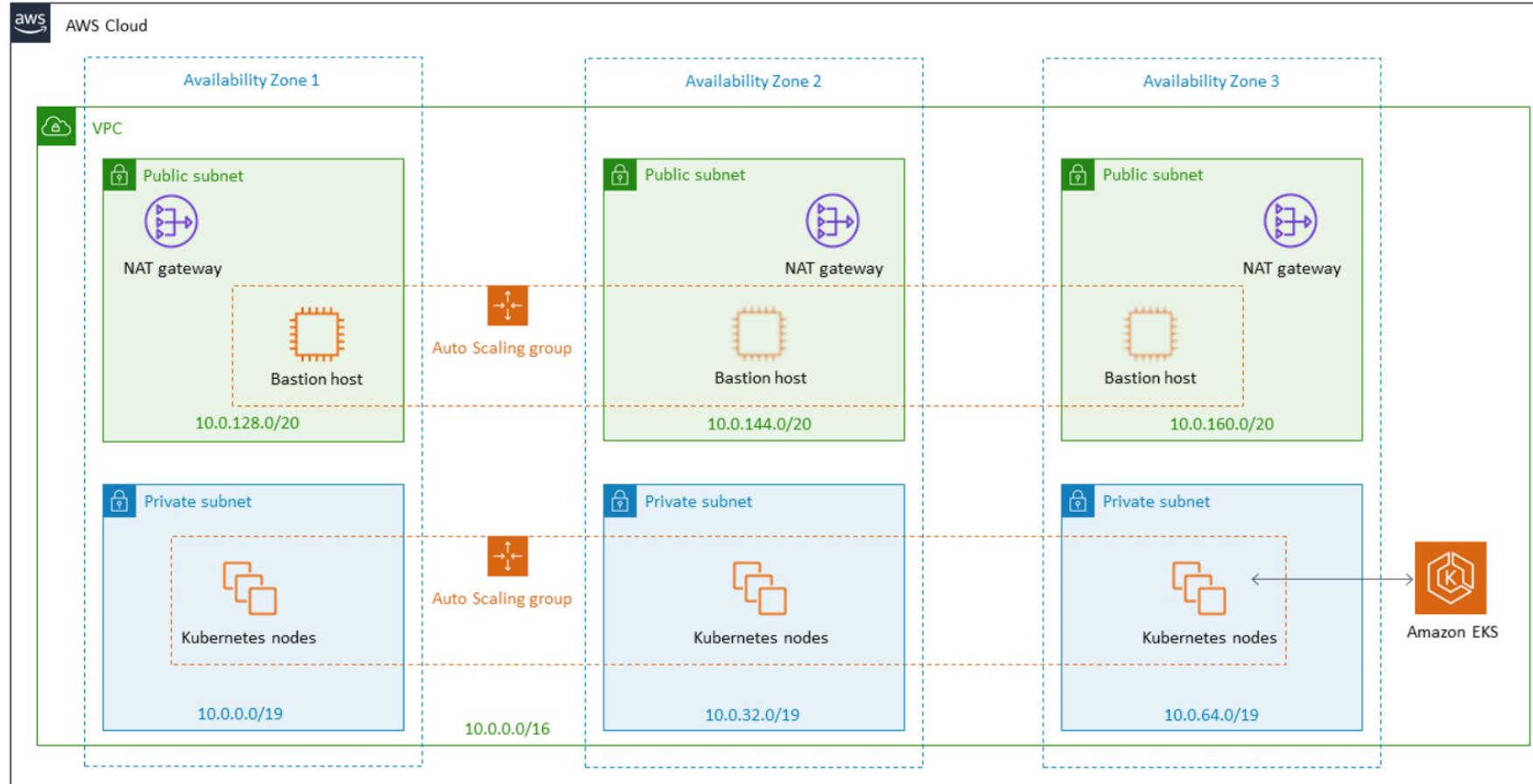
a. Fargate – creates the serverless containers.

b. Amazon EC2's – Deploy the Worker nodes and map to the EKS Cluster.



Architecture - Deployment Diagram:

Details are elaborated in the next slide.

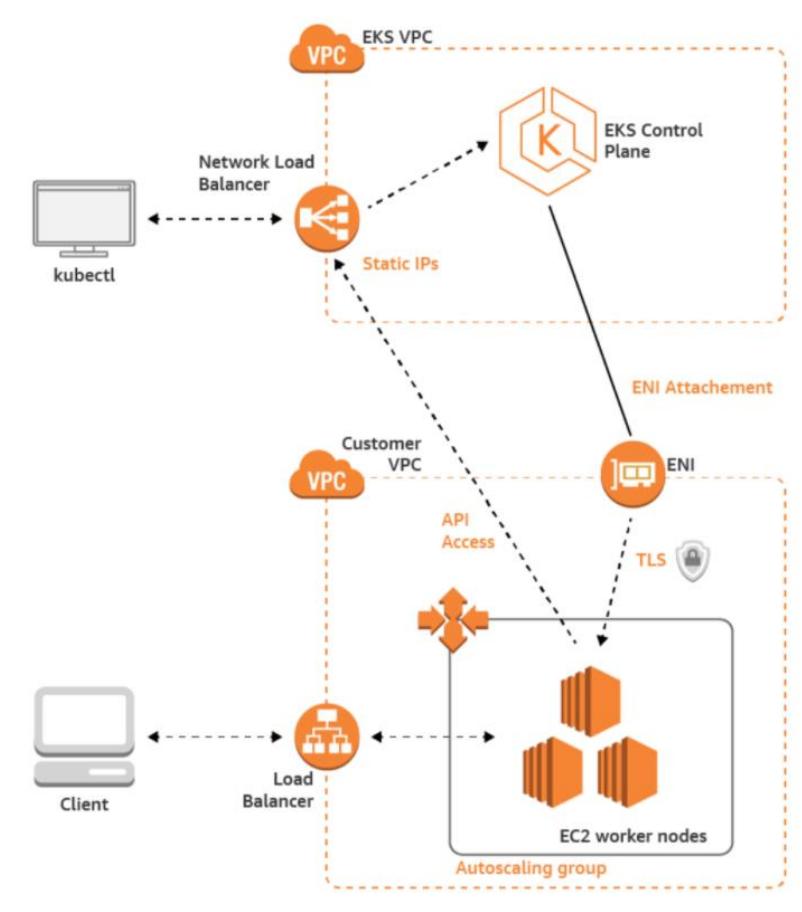


Architecture - Deployment Diagram - Description:

- A highly available architecture that spans three Availability Zones.
- A VPC, virtual private cloud is configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.
- Managed NAT gateways in the public subnets will allow outbound internet access for resources in the private subnets.
- In one public subnet, a Linux bastion host in an Auto Scaling group to allow inbound Secure Shell (SSH) access to Amazon Elastic Compute Cloud (Amazon EC2) instances in private subnets. The bastion host is also configured with the Kubernetes kubectl command line interface for managing the Kubernetes cluster.
- An Amazon EKS cluster, which provides the Kubernetes control plane.
- In the private subnets, a group of Kubernetes Worker nodes are deployed through Auto Scaling group.

Architecture – Activity diagram:

Details are elaborated in the next slide.



Architecture - Activity diagram - Description:

1. Client will communicate through the Load Balancer and the request goes to Worker nodes.
2. Worker nodes communicates with EKS Control Plane, formed by EKS cluster, through the Network Load Balancer.
3. EKS Control Plane, communicates with the Worker nodes through ENI.
4. Kubectl CLI provides CLI access to EKS Control Plane, in-turn even controls the Worker nodes.
5. Worker nodes are configured with a Autoscaling group.

Details of the Case Study/ POC:

- Deploying an nginx sample application on the Amazon EKS Cluster.
- There will be an EKS cluster created on the AWS with the Worker nodes connected, which are configured with Auto scaling.

Steps, Hands-on activity:

1. Create Role – Create IAM role with necessary policies attached to make the user to access EKS services.
2. Create VPC – Through a CloudFormation template, to create the necessary secured networking for the EKS services.
3. Create EKS Cluster.
4. Install Kubectl – Used to access EKS cluster through CLI.
5. Install Eksctl – Enables the users to access the EKS services through CLI.
6. Install aws-iam-authenticator - Enables the users to authenticate and access the EKS Cluster through CLI from the remote systems.
7. Create Worker nodes.
8. Map the worker nodes with EKS Cluster.
9. Deploy the Sample application.
10. Delete Worker nodes and the EKS Cluster.

Create Role:

The screenshot shows the AWS IAM Roles page. The left sidebar has a 'Roles' section selected. The main area displays a table of existing roles, a search bar, and a 'Create role' button.

Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Role name	Trusted entities	Last activity
AWSCodePipelineServiceRole-us-...	AWS service: codepipeline	153 days
AWSServiceRoleForAmazonCode...	AWS service: codeguru-reviewer (Service-Link...	116 days
AWSServiceRoleForAmazonEKS	AWS service: eks (Service-Linked role)	None

Choose the EKS service:

The screenshot shows the AWS IAM Role Creation Wizard at the 'Choose the EKS service' step. The URL in the browser is `console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=type`. The top navigation bar includes a star icon, a refresh icon, a camera icon, and a search icon. The user is signed in as 'SubbuAWSCG'. The main content area displays four options for choosing a service type:

- AWS service**: EC2, Lambda and others
- Another AWS account**: Belonging to you or 3rd party
- Web identity**: Cognito or any OpenID provider
- SAML 2.0 federation**: Your corporate directory

Below these options, a note states: "Allows AWS services to perform actions on your behalf. [Learn more](#)".

Choose a use case

Common use cases

- EC2**: Allows EC2 instances to call AWS services on your behalf.
- Lambda**: Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CloudWatch Events	EKS	KMS	Rekognition
AWS Backup	CodeBuild	EMR	Kinesis	RoboMaker
AWS Chatbot	CodeDeploy	ElastiCache	Lake Formation	S3

* Required

Cancel

Next: Permissions

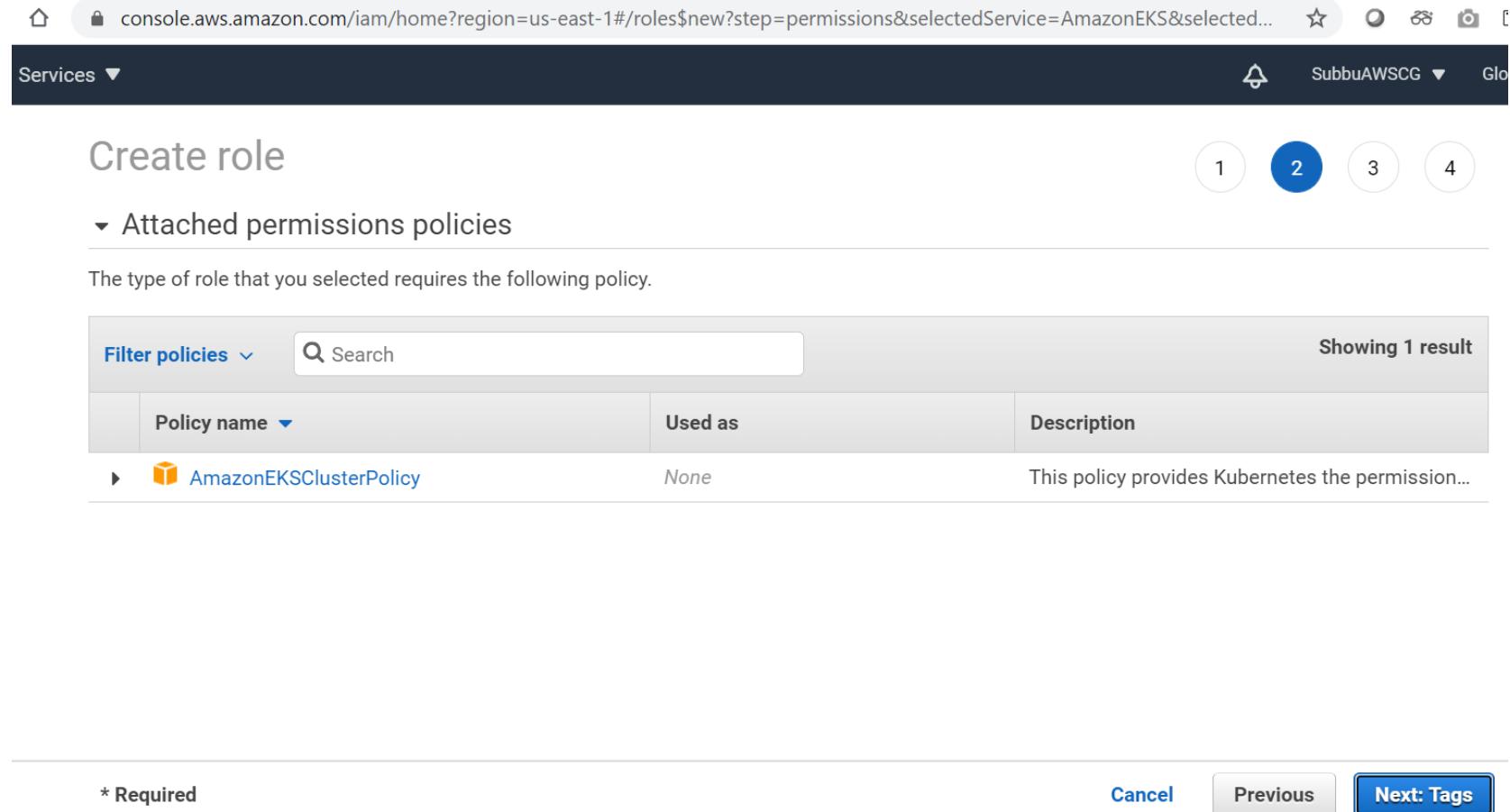
Choose the appropriate use case:

The screenshot shows a browser window for the AWS IAM console at `console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=type`. The AWS logo is in the top left, and the user 'SubbuAWSCG' is logged in. The main content area is titled 'Select your use case' and lists several options under the heading 'EKS':

- EKS**: Allows EKS to manage clusters on your behalf.
- EKS - Cluster**: Allows access to other AWS service resources that are required to operate clusters managed by EKS. This option is highlighted with a light blue background.
- 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**: Allows EKS to manage nodegroups on your behalf.

At the bottom, there is a note: '* Required'. On the right, there are 'Cancel' and 'Next: Permissions' buttons.

Attach the Policy: AmazonEKSClusterPolicy.

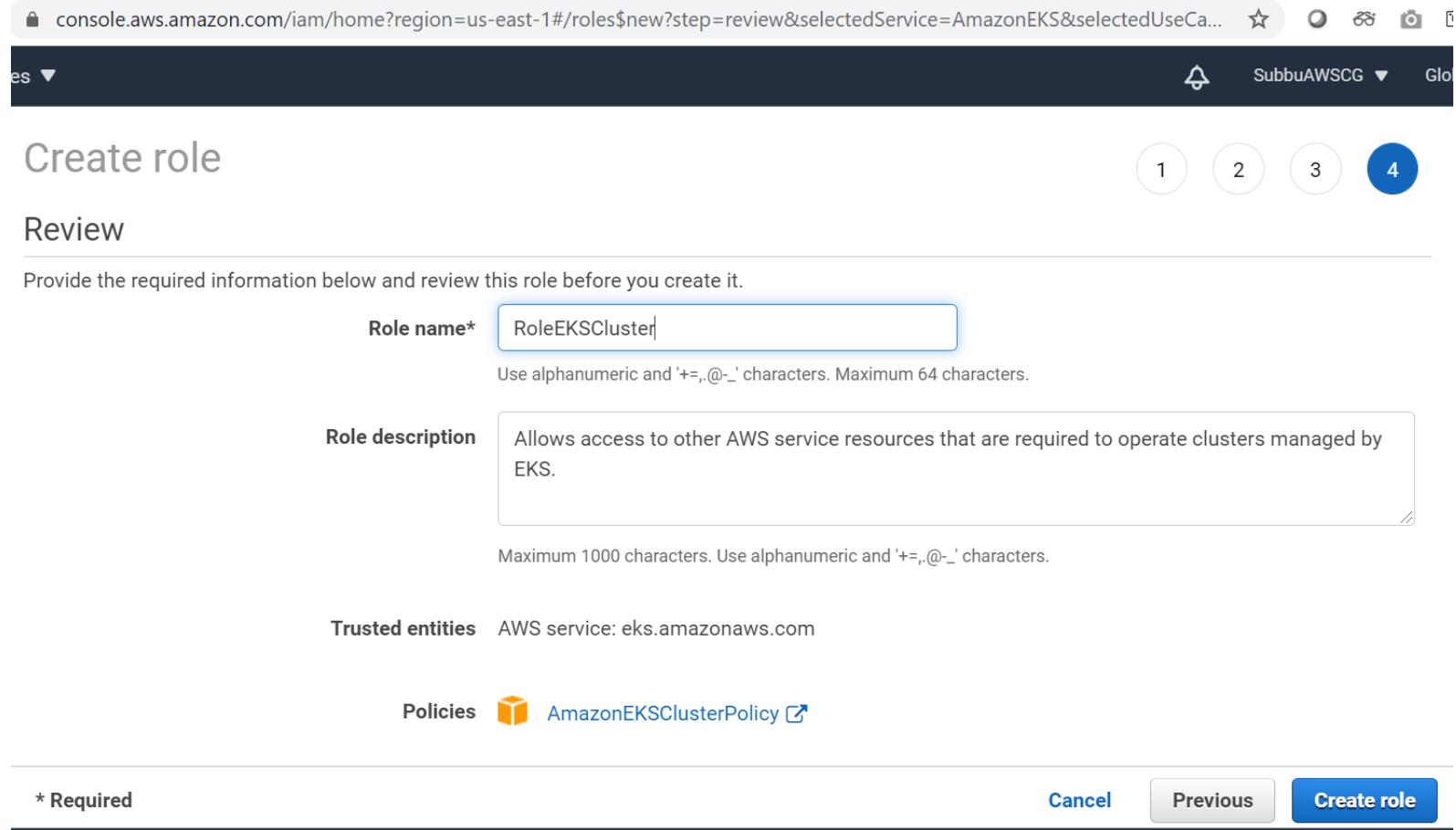


The screenshot shows the AWS IAM 'Create role' wizard at step 2. The URL in the browser is `console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=permissions&selectedService=AmazonEKS&selected...`. The page title is 'Create role'. Step 2 is highlighted with a blue circle. Below the title, there's a section titled 'Attached permissions policies' with a dropdown arrow. A note says 'The type of role that you selected requires the following policy.' A table lists one policy:

Policy name	Used as	Description
▶ AmazonEKSClusterPolicy	None	This policy provides Kubernetes the permission...

At the bottom, there are buttons for 'Cancel', 'Previous', and 'Next: Tags'.

Give the Role name:



The screenshot shows the 'Create role' review step in the AWS IAM console. The URL in the address bar is `console.aws.amazon.com/iam/home?region=us-east-1#/roles$new?step=review&selectedService=AmazonEKS&selectedUseCa...`. The top navigation bar includes links for Home, Services, SubbuAWSRG, and Global. Below the navigation, the page title is 'Create role' and the sub-section is 'Review'. A progress bar at the top right shows four steps, with step 4 being the active one. The main content area contains fields for 'Role name*' (set to 'RoleEKSCluster') and 'Role description' (set to 'Allows access to other AWS service resources that are required to operate clusters managed by EKS'). Below these fields, a note states 'Maximum 1000 characters. Use alphanumeric and '+,-,@,_' characters.' Under 'Trusted entities', it says 'AWS service: eks.amazonaws.com'. At the bottom, there are buttons for 'Policies' (with a yellow cube icon), 'AmazonEKSClusterPolicy' (with a blue arrow icon), 'Cancel', 'Previous', and a prominent 'Create role' button.

console.aws.amazon.com/iam/home?region=us-east-1#/roles\$new?step=review&selectedService=AmazonEKS&selectedUseCa...

SubbuAWSRG ▾ Global

Create role

Review

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

Role name* RoleEKSCluster

Use alphanumeric and '+,-,@,_' characters. Maximum 64 characters.

Role description Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Maximum 1000 characters. Use alphanumeric and '+,-,@,_' characters.

Trusted entities AWS service: eks.amazonaws.com

Policies  AmazonEKSClusterPolicy 

* Required

Cancel Previous Create role

Create VPC: using CloudFormation template.

The screenshot shows the AWS CloudFormation console home page. The URL in the browser is `console.aws.amazon.com/cloudformation/home?region=us-east-1#/`. The page features a dark header with the AWS logo, a 'Services' dropdown, and user account information ('SubbuAWSCG' and 'N. Virginia'). Below the header, there's a navigation bar with 'Management & Governance' and a search bar. The main content area has a dark background with white text. It features the 'AWS CloudFormation' logo and the tagline 'Model and provision all your cloud infrastructure'. A call-to-action button labeled 'Create stack' is visible. To the left, a section titled 'How it works' includes a 'Simplify Your Infrastructure Management U...' link. On the right, there are sections for 'Getting started' (with links to 'What is AWS CloudFormation' and 'Getting started with CloudFormation') and 'CloudFormation resources' (with links to 'AWS Lambda', 'Amazon S3', 'Amazon RDS', and 'Amazon VPC').

Choose the ready template from S3:

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The top navigation bar includes the AWS logo, 'Services ▾', a bell icon, 'SubbuAWSCG ▾', 'N. Virginia ▾', and 'Support ▾'. The left sidebar lists steps: Step 1 'Specify template' (selected), Step 2 'Specify stack details', Step 3 'Configure stack options', and Step 4 'Review'. The main content area is titled 'Create stack' and 'Prerequisite - Prepare template'. It explains that every stack is based on a template, which is a JSON or YAML file containing configuration information for AWS resources. Three radio button options are shown: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. Below this is the 'Specify template' section, which defines a template as a JSON or YAML file describing stack resources and properties. It includes a 'Template source' section for selecting a template URL, with 'Amazon S3 URL' selected (radio button checked) and 'Upload a template file' as an alternative. An input field contains the URL `https://`. The 'Amazon S3 template URL' label is below it. At the bottom of the 'Specify template' section is the note 'S3 URL: Will be generated when URL is provided' and a 'View in Designer' button. The bottom navigation bar features 'Cancel' and 'Next' buttons, along with links for 'Feedback', 'English (US) ▾', '© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

Get the S3 URL from the AWS documentation in the below link:

<https://docs.aws.amazon.com/eks/latest/userguide/create-public-private-vpc.html>

<https://amazon-eks.s3.us-west-2.amazonaws.com/clouformation/2020-08-12/amazon-eks-vpc-private-subnets.yaml>

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The left sidebar lists steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main area is titled 'Create stack' and 'Prerequisite - Prepare template'. It explains that every stack is based on a template, which is a JSON or YAML file containing configuration information for AWS resources. Three options are shown: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. Below this is the 'Specify template' section, which asks for the 'Template source'. It says selecting a template generates an Amazon S3 URL where it will be stored. Two options are shown: 'Amazon S3 URL' (selected) and 'Upload a template file'. An input field contains the URL: <https://amazon-eks.s3.us-west-2.amazonaws.com/clouformation/2020-08-12/amazon-eks-vpc-private-subnets.yaml>. A note below it says 'Amazon S3 template URL'. At the bottom right of this section is a 'View in Designer' button. At the very bottom are 'Cancel' and 'Next' buttons.

Give a stack name and the predefined VPC configurations will get loaded from S3:

The screenshot shows the 'Specify stack details' step of the AWS CloudFormation 'Create stack' wizard. The left sidebar lists steps: Step 1 (Specify template), Step 2 (Specify stack details, currently selected), Step 3 (Configure stack options), and Step 4 (Review). The main area is titled 'Specify stack details'.

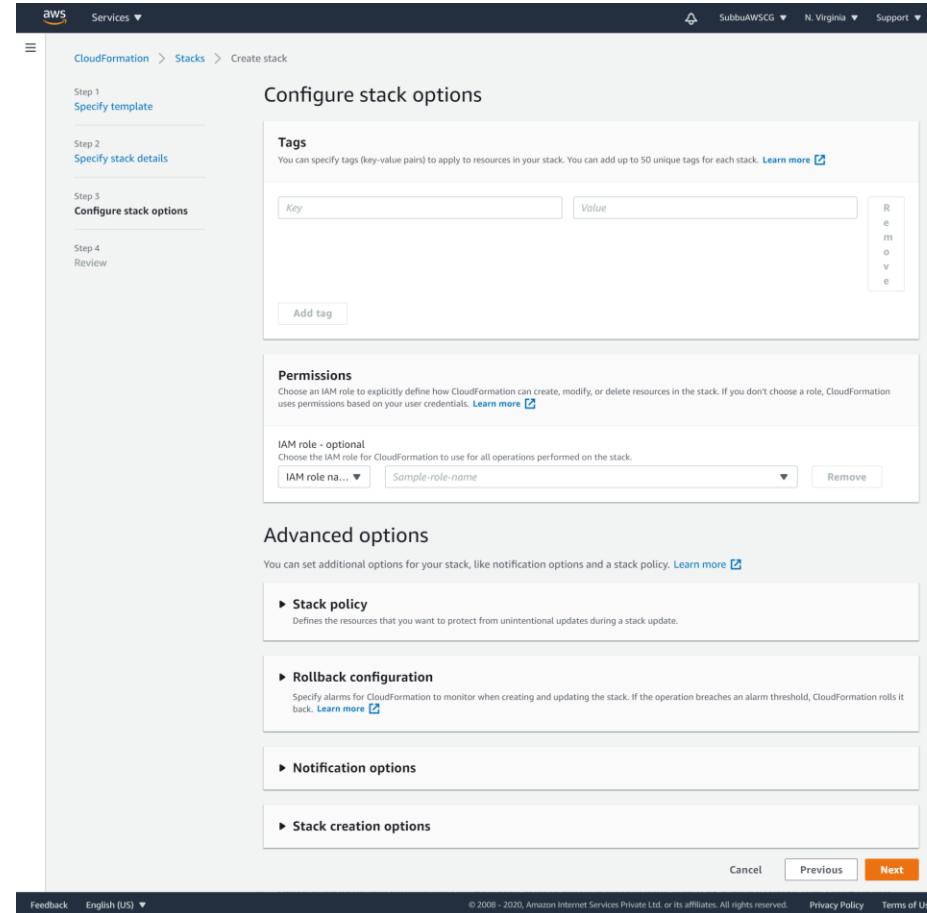
Stack name: CloudFormationVPC4EKS

Parameters:

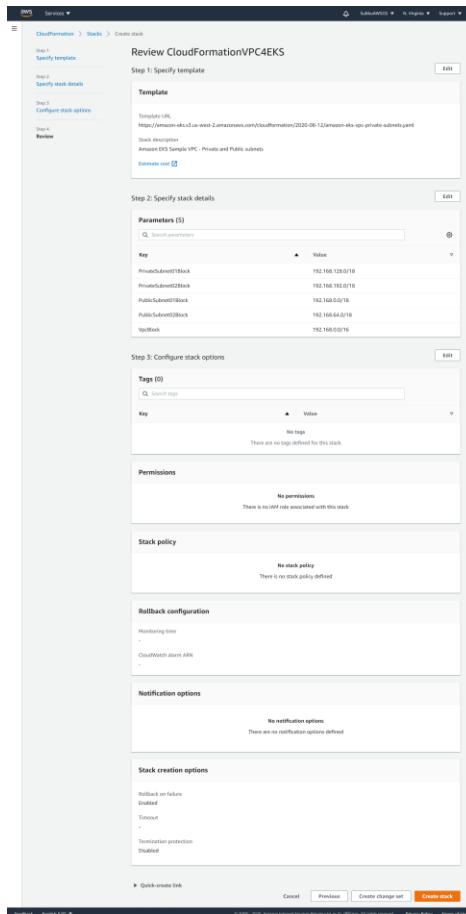
- VpcBlock:** 192.168.0.0/16
- PublicSubnet01Block:** 192.168.0.0/18
- PublicSubnet02Block:** 192.168.64.0/18
- PrivateSubnet01Block:** 192.168.128.0/18
- PrivateSubnet02Block:** 192.168.192.0/18

At the bottom are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' highlighted in orange.

Input more advanced options if needed:



Review VPC details:



VPC creation through CloudFormation is in progress:

The screenshot shows the AWS CloudFormation console interface. On the left, a sidebar lists several stacks, including 'CloudFormationVPC4EKS' which is highlighted. The main area displays the details for this stack. At the top right, there are buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'. Below these are tabs for 'Stack info', 'Events' (which is selected), 'Resources', 'Outputs', 'Parameters', 'Template', and 'Change sets'. The 'Events' tab shows one event entry:

Timestamp	Logical ID	Status	Status reason
2020-10-07 12:42:27 UTC+0530	CloudFormationVPC4EKS	CREATE_IN_PROGRESS	User Initiated

At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Privacy Policy'.

Detailed logs of VPC creation through CloudFormation:



VPC create is complete through CloudFormation:

The screenshot shows the AWS CloudFormation Stacks page with the following details:

Stack name	Status	Created time	Description
CloudFormationVPC4EKS	CREATE_COMPLETE	2020-10-07 12:42:27 UTC+0530	Amazon EKS Sample VPC - Private and ...
sam-app	ROLLBACK_COMPLETE	2020-09-25 12:50:10 UTC+0530	Blockchain application with Amazon M...
aws-sam-cli-managed-default	CREATE_COMPLETE	2020-09-25 12:48:41 UTC+0530	Managed Stack for AWS SAM CLI
aws-cloud9-SubbuAWSC9Environment...	CREATE_COMPLETE	2020-09-22 13:36:18 UTC+0530	-
AWSBlockchainTemplateforHyperledg...	ROLLBACK_COMPLETE	2020-07-09 15:00:42 UTC+0530	AWS Blockchain Template for Hyperled...
EC2ContainerService-SubbuECSCluster	CREATE_COMPLETE	2020-06-19 16:21:49 UTC+0530	AWS CloudFormation template to creat...

Page navigation and settings are visible at the top and bottom of the interface.

The view of VPC CloudFormation output:

The screenshot shows the AWS CloudFormation Outputs page for the stack named "CloudFormationVPC4EKS".

Stacks (6) sidebar:

- CloudFormationVPC4EKS (Active, CREATE_COMPLETE)
- sam-app (Incomplete, ROLLBACK_COMPLETE)
- aws-sam-cli-managed-default (Incomplete, CREATE_COMPLETE)
- aws-cloud9-SubbuAWS9Environment-0b0542c98eb94707a53c671e40b4979b (Incomplete, CREATE_COMPLETE)
- AWSBlockchainTemplateforHyperledgerFabricCF (Incomplete, ROLLBACK_COMPLETE)
- EC2ContainerService-SubbuECSCluster (Incomplete, CREATE_COMPLETE)

CloudFormationVPC4EKS main page:

Outputs (3) tab is selected.

Key	Value	Description	Export name
Security group for the cluster control plane	(No value shown)	Security group for the cluster control plane	(No export name shown)

Page footer:

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

Created VPC:

The screenshot shows the AWS VPC console interface. The top navigation bar includes links for back, forward, home, and search, along with the URL `console.aws.amazon.com/vpc/home?region=us-east-1#vpcs:`. The header also displays the AWS logo, user SubbuAWSCG, region N. Virginia, and support links.

The left sidebar features a "New VPC Experience" survey, "Site-to-Site VPN Connections", "Client VPN Endpoints", and a "TRANSIT GATEWAYS" section with "Transit Gateways" and "Transit Gateway Attachments".

The main content area is titled "Your VPCs (1/3)" and contains a table with the following data:

	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	SubbuDefaultVPC	vpc-081bf7a2e5bb4ebdd	Available	172.31.0.0/16
<input type="checkbox"/>	SubbuVPC	vpc-0334ac2a8f318c2bc	Available	10.0.0.0/24
<input checked="" type="checkbox"/>	CloudFormationVPC4EKS-VPC	vpc-0a4c191b5b728de1c	Available	192.168.0.0/16

Details of created VPC:

The screenshot shows the AWS VPC Details page for a specific VPC. The VPC ID is `vpc-0a4c191b5b728de1c`, and the name is `CloudFormationVPC4EKS-VPC`. The page displays various configuration details such as State (Available), DNS hostnames (Enabled), and Network ACL (acl-0fe9f777baa86d973). The VPC has a CIDR range of `192.168.0.0/16`.

Details		Info	
VPC ID	 vpc-0a4c191b5b728de1c	State	Available
Tenancy		DHCP options set	
Default		dopt-f3c17589	Route table rtb-01d1c14497f1b1f65
Default VPC		IPv4 CIDR	IPv6 pool -
No		192.168.0.0/16	IPv6 CIDR (Network Border Group) -
Owner ID	 112253241392		

Route table, that automatically comes with the VPC:

The screenshot shows the AWS VPC Route Tables page. The URL in the browser is `console.aws.amazon.com/vpc/home?region=us-east-1#RouteTables:routeTableId=rtb-01d1c14497f1b1f65;sort=routeTableId`. The page title is "Create route table". A search bar at the top right contains "Route Table ID : rtb-01d1c14497f1b1f65" and an "Add filter" button. Below the search is a table with one row, highlighted with a yellow background. The table columns are Name, Route Table ID, Explicit subnet association, Edge associations, Main, and VPC ID. The row shows "rtb-01d1c14497f1b1f65" in the Route Table ID column, and "Yes" in the Main column. The VPC ID is "vpc-0a4c191b5b7". On the left sidebar, under "VIRTUAL PRIVATE CLOUD", the "Route Tables" section is selected. At the bottom of the page, there are tabs for Summary, Routes, Subnet Associations, Edge Associations, Route Propagation, and Tags. The "Summary" tab is active. Below the tabs, detailed information is provided: Route Table ID (rtb-01d1c14497f1b1f65), Main (Yes), VPC (vpc-0a4c191b5b728de1c | CloudFormationVPC4EKS-VPC), Explicitly Associated with (-), and Owner (112253241392).

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
	rtb-01d1c14497f1b1f65	-	-	Yes	vpc-0a4c191b5b7

Route Table: rtb-01d1c14497f1b1f65

Route Table ID: rtb-01d1c14497f1b1f65

Main: Yes

VPC: vpc-0a4c191b5b728de1c | CloudFormationVPC4EKS-VPC

Owner: 112253241392

Details of Network ACL (Access Control List) associated with VPC:

The screenshot shows the AWS VPC Network ACL details page. The URL in the address bar is `aws.amazon.com/vpc/home?region=us-east-1#acls:networkAclId=acl-0fe9f777baa86d973;sort=networkAclId`. The top navigation bar includes links for SubbuAWSCG, N. Virginia, and Support.

The main content area displays a table of Network ACLs. A search bar at the top of the table lists "Network ACL ID : acl-0fe9f777baa86d973". The table columns are Name, Network ACL ID, Associated with, Default, VPC, and Owner. One row is visible, showing "acl-0fe9f777baa86..." associated with "4 Subnets", marked as "Default", and linked to "vpc-0a4c191b5b728de1c | CloudFormatio...".

Below the table, tabs for Details, Inbound Rules, Outbound Rules, Subnet associations, and Tags are present. The "Subnet associations" tab is selected, showing a list of subnets with their IPv4 CIDR ranges. The subnets listed are:

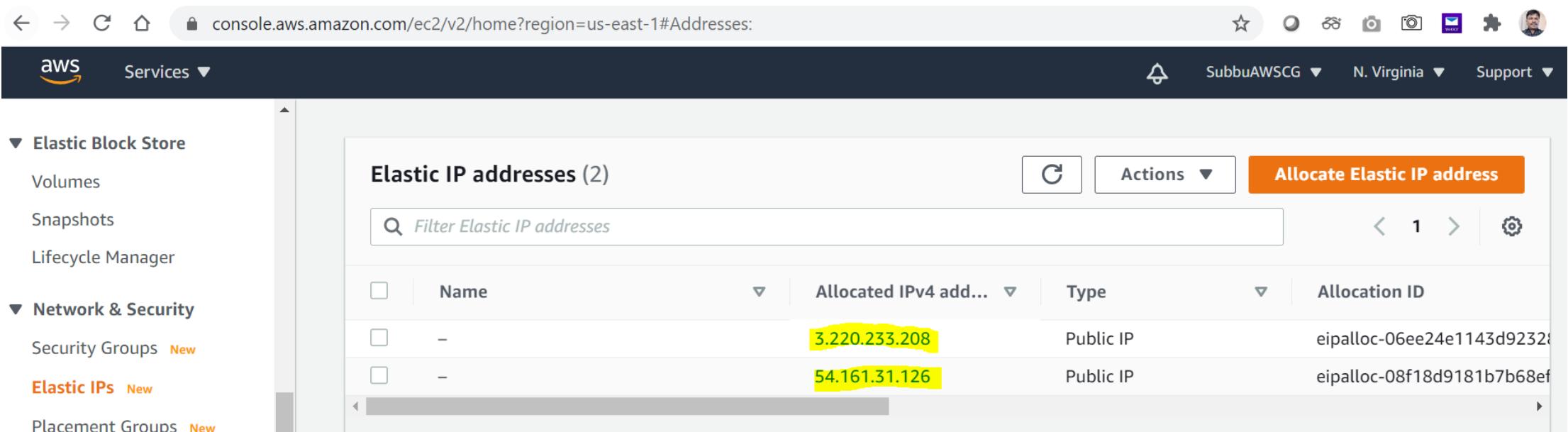
Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0830f0b2b8...	192.168.0.0/18	-
subnet-05644000bf...	192.168.128.0/18	-
subnet-0cc66aaa0bb...	192.168.192.0/18	-
subnet-0fd725c78b...	192.168.64.0/18	-

Created VPC from the CloudFront:

The screenshot shows the AWS VPC Details page for a specific VPC. The URL in the browser is `console.aws.amazon.com/vpc/home?region=us-east-1#VpcDetails:VpcId=vpc-0a4c191b5b728de1c`. The page title is `vpc-0a4c191b5b728de1c / CloudFormationVPC4EKS-VPC`. The main content area displays the VPC's details in a tabular format.

Details		Actions	
VPC ID	vpc-0a4c191b5b728de1c	State	Available
Tenancy		DHCP options set	dopt-f3c17589
Default		Route table	rtb-01d1c14497f1b1f65
Default VPC		IPv6 pool	
No		-	IPv6 CIDR (Network Border Group)
Owner ID	112253241392		-

Elastic IP addresses' created:



The screenshot shows the AWS Management Console interface for managing Elastic IP addresses. The URL in the browser is `console.aws.amazon.com/ec2/v2/home?region=us-east-1#Addresses`. The top navigation bar includes links for Services, SubbuAWSCG, N. Virginia, and Support. On the left, a sidebar menu is open under the Network & Security section, showing options for Security Groups, Elastic IPs (which is currently selected), and Placement Groups.

The main content area is titled "Elastic IP addresses (2)". It features a search bar labeled "Filter Elastic IP addresses" and a table with the following data:

<input type="checkbox"/>	Name	Allocated IPv4 add...	Type	Allocation ID
<input type="checkbox"/>	-	3.220.233.208	Public IP	eipalloc-06ee24e1143d92328
<input type="checkbox"/>	-	54.161.31.126	Public IP	eipalloc-08f18d9181b7b68ef

Details of the EIP's:

The screenshot shows the AWS EC2 Elastic IP addresses page. The top navigation bar includes the AWS logo, Services dropdown, SubbuAWSCG account, N. Virginia region, and Support link. The main navigation path is EC2 > Elastic IP addresses > 3.220.233.208. The page title is 3.220.233.208. There are two tabs: Actions and Associate Elastic IP address. The Summary section displays the following details:

Allocated IPv4 address	Type	Allocation ID	Association ID
3.220.233.208	Public IP	eipalloc-06ee24e1143d92328	eipassoc-00f2fce91fd6737d3
Scope	Associated instance ID	Private IP address	Network interface ID
VPC	-	192.168.31.105	eni-099ba7fa423e3b3df
Network interface owner account ID	Public DNS	NAT Gateway ID	Address pool
112253241392	ec2-3-220-233-208.compute-1.amazonaws.com	nat-0eccdbcdea7159168 (CloudFormationVPC4EKS-NatGatewayA21)	Amazon
Network Border Group	us-east-1		

The Tags section shows three tags:

Key	Value
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:112253241392:stack/CloudFormationVPC4EKS/74c8f7f0-086c-11eb-9085-12548a4acf7
aws:cloudformation:logical-id	NatGatewayEIP1
aws:cloudformation:stack-name	CloudFormationVPC4EKS

At the bottom, there are links for Feedback, English (US), Copyright notice (2008-2020), Privacy Policy, and Terms of Use.

Details of the EIP's:

The screenshot shows the AWS EC2 Elastic IP addresses details page for the IP address 54.161.31.126. The page has a header with the AWS logo, Services dropdown, SubbuAWSCG, N. Virginia, and Support. The main content area has a summary table and a tags section.

Summary

Allocated IPv4 address 54.161.31.126	Type Public IP	Allocation ID eipalloc-08f18d9181b7b68ef	Association ID eipassoc-08cc18e5589d49ab4
Scope VPC	Associated instance ID -	Private IP address 192.168.111.19	Network interface ID eni-0898ac5d29db8cfa7
Network interface owner account ID 112253241392	Public DNS ec2-54-161-31-126.compute-1.amazonaws.com	NAT Gateway ID nat-02e4ebc799ef88af7 (CloudFormationVPC4EKS-NatGatewayA22)	Address pool Amazon
Network Border Group us-east-1			

Tags (3)

Key	Value
aws:cloudformation:stack-name	CloudFormationVPC4EKS
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:112253241392:stack/CloudFormationVPC4EKS/74c8f7f0-086c-11eb-9085-1254a8a4acf7
aws:cloudformation:logical-id	NatGatewayEIP2

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

Security Group, formed along with the VPC – Inbound:

The screenshot shows the AWS EC2 Security Groups details page for a security group named "sg-0952d90b22ed21a6d - SubbuEKSCluster-worker-nodes-NodeSecurityGroup-KZNI8RCAGDVE".

Details:

Security group name SubbuEKSCluster-worker-nodes-NodeSecurityGroup-KZNI8RCAGDVE	Security group ID sg-0952d90b22ed21a6d	Description Security group for all nodes in the cluster	VPC ID vpc-0a4c191b5b728de1c
Owner 112253241392	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules:

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	sg-0952d90b22ed21a6d (SubbuEKSCluster-worker-nodes-NodeSecurityGroup-KZNI8RCAGDVE)	Allow node to communicate with each other
Custom TCP	TCP	1025 - 65535	sg-051fa037dce80c2ec (CloudFormationVPC4EKS-ControlPlaneSecurityGroup-1D5HKGKV5WO4lQ2)	Allow worker Kubelets and pods to receive communication from the cluster control plane
HTTPS	TCP	443	sg-051fa037dce80c2ec (CloudFormationVPC4EKS-ControlPlaneSecurityGroup-1D5HKGKV5WO4lQ2)	Allow pods running extension API servers on port 443 to receive communication from cluster control plane

Security Group, formed along with the VPC – Outbound:

The screenshot shows the AWS EC2 Security Groups console. The top navigation bar includes the AWS logo, Services dropdown, SubbuAWSCG profile, N. Virginia region, and Support links. The main title is "sg-0952d90b22ed21a6d - SubbuEKSCluster-worker-nodes-NodeSecurityGroup-KZNI8RCAGDVE". Below the title, there's a "Details" section with the following information:

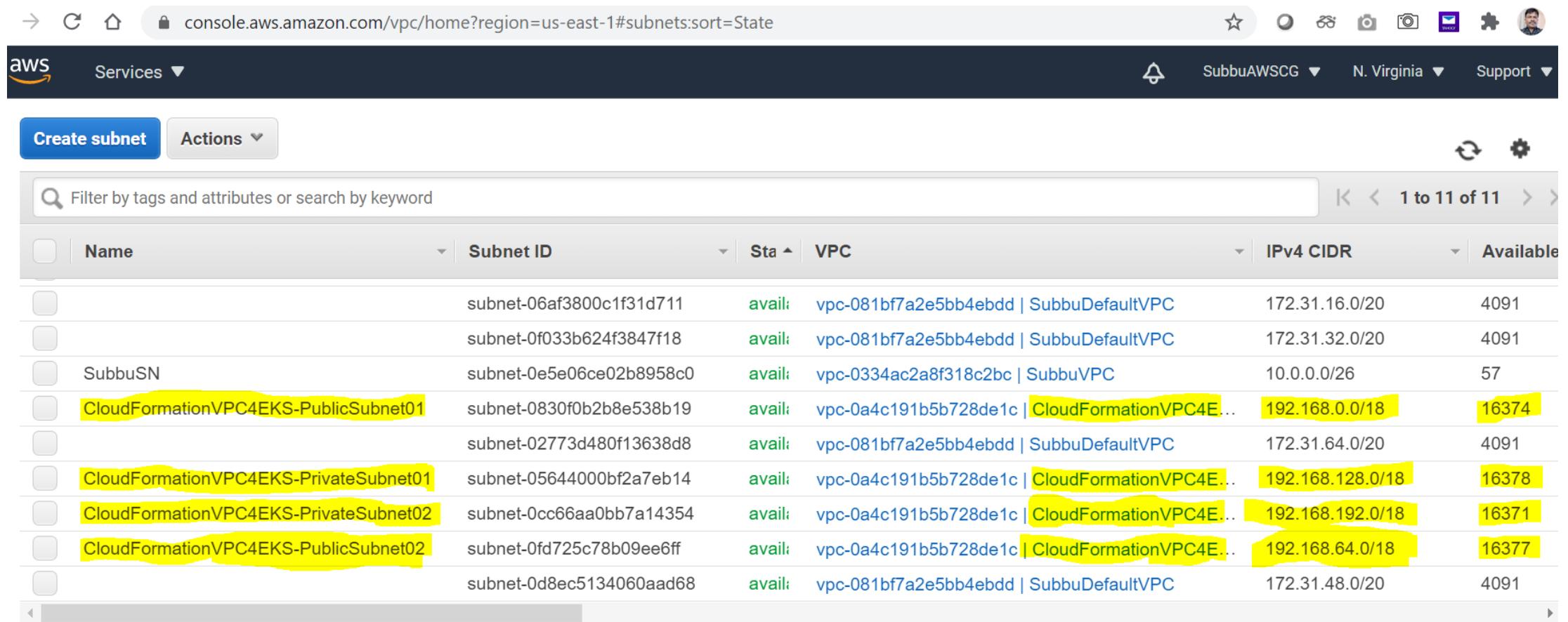
Security group name	sg-0952d90b22ed21a6d	Description	VPC ID
Owner	SubbuEKSCluster-worker-nodes-NodeSecurityGroup-KZNI8RCAGDVE	Inbound rules count	vpc-0a4c191b5b728de1c
	112253241392	3 Permission entries	1 Permission entry

Below the details, there are tabs for "Inbound rules", "Outbound rules" (which is selected), and "Tags". The "Outbound rules" section contains one rule:

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

At the bottom, there are links for Feedback, English (US) dropdown, Copyright notice (© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

Created Subnet's, as a part of VPC with CloudFormation execution:



console.aws.amazon.com/vpc/home?region=us-east-1#subnets:sort=State

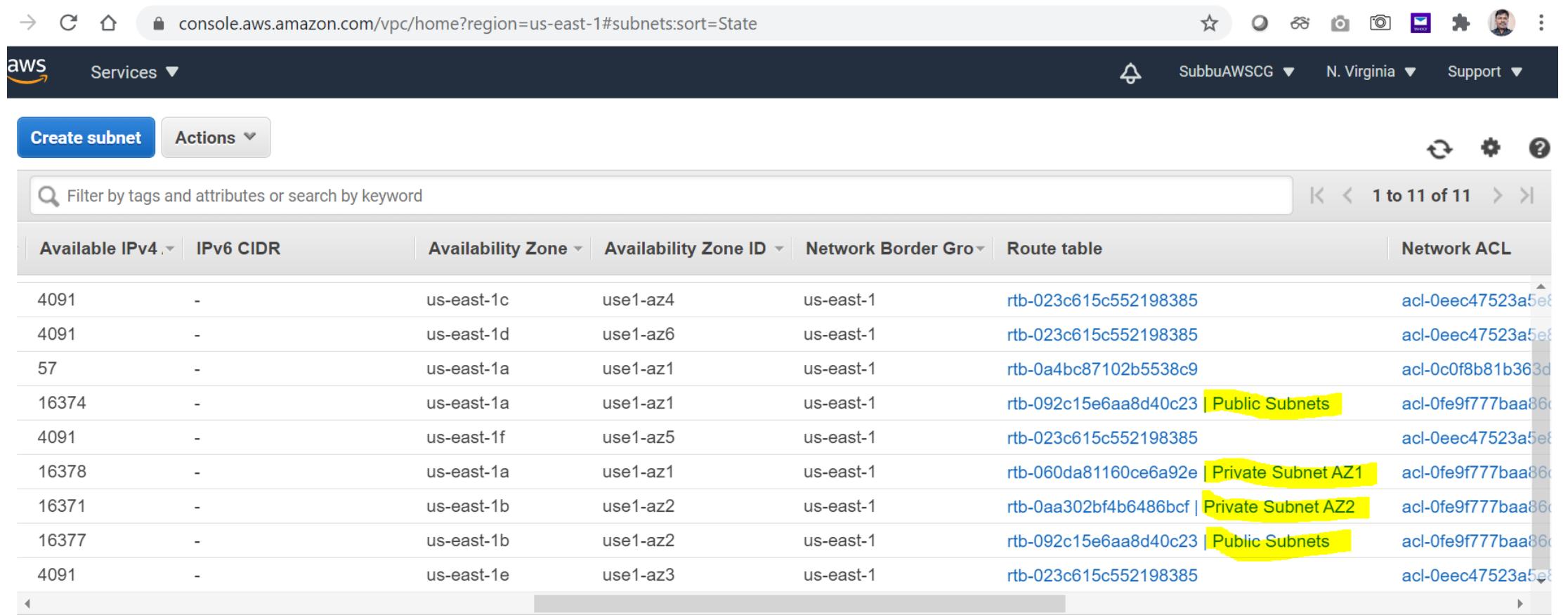
Create subnet Actions ▾

Filter by tags and attributes or search by keyword

1 to 11 of 11

<input type="checkbox"/>	Name	Subnet ID	Status	VPC	IPv4 CIDR	Available
<input type="checkbox"/>		subnet-06af3800c1f31d711	avail	vpc-081bf7a2e5bb4ebdd SubbuDefaultVPC	172.31.16.0/20	4091
<input type="checkbox"/>		subnet-0f033b624f3847f18	avail	vpc-081bf7a2e5bb4ebdd SubbuDefaultVPC	172.31.32.0/20	4091
<input type="checkbox"/>	SubbuSN	subnet-0e5e06ce02b8958c0	avail	vpc-0334ac2a8f318c2bc SubbuVPC	10.0.0.0/26	57
<input type="checkbox"/>	CloudFormationVPC4EKS-PublicSubnet01	subnet-0830f0b2b8e538b19	avail	vpc-0a4c191b5b728de1c CloudFormationVPC4E...	192.168.0.0/18	16374
<input type="checkbox"/>		subnet-02773d480f13638d8	avail	vpc-081bf7a2e5bb4ebdd SubbuDefaultVPC	172.31.64.0/20	4091
<input type="checkbox"/>	CloudFormationVPC4EKS-PrivateSubnet01	subnet-05644000bf2a7eb14	avail	vpc-0a4c191b5b728de1c CloudFormationVPC4E...	192.168.128.0/18	16378
<input type="checkbox"/>	CloudFormationVPC4EKS-PrivateSubnet02	subnet-0cc66aa0bb7a14354	avail	vpc-0a4c191b5b728de1c CloudFormationVPC4E...	192.168.192.0/18	16371
<input type="checkbox"/>	CloudFormationVPC4EKS-PublicSubnet02	subnet-0fd725c78b09ee6ff	avail	vpc-0a4c191b5b728de1c CloudFormationVPC4E...	192.168.64.0/18	16377
<input type="checkbox"/>		subnet-0d8ec5134060aad68	avail	vpc-081bf7a2e5bb4ebdd SubbuDefaultVPC	172.31.48.0/20	4091

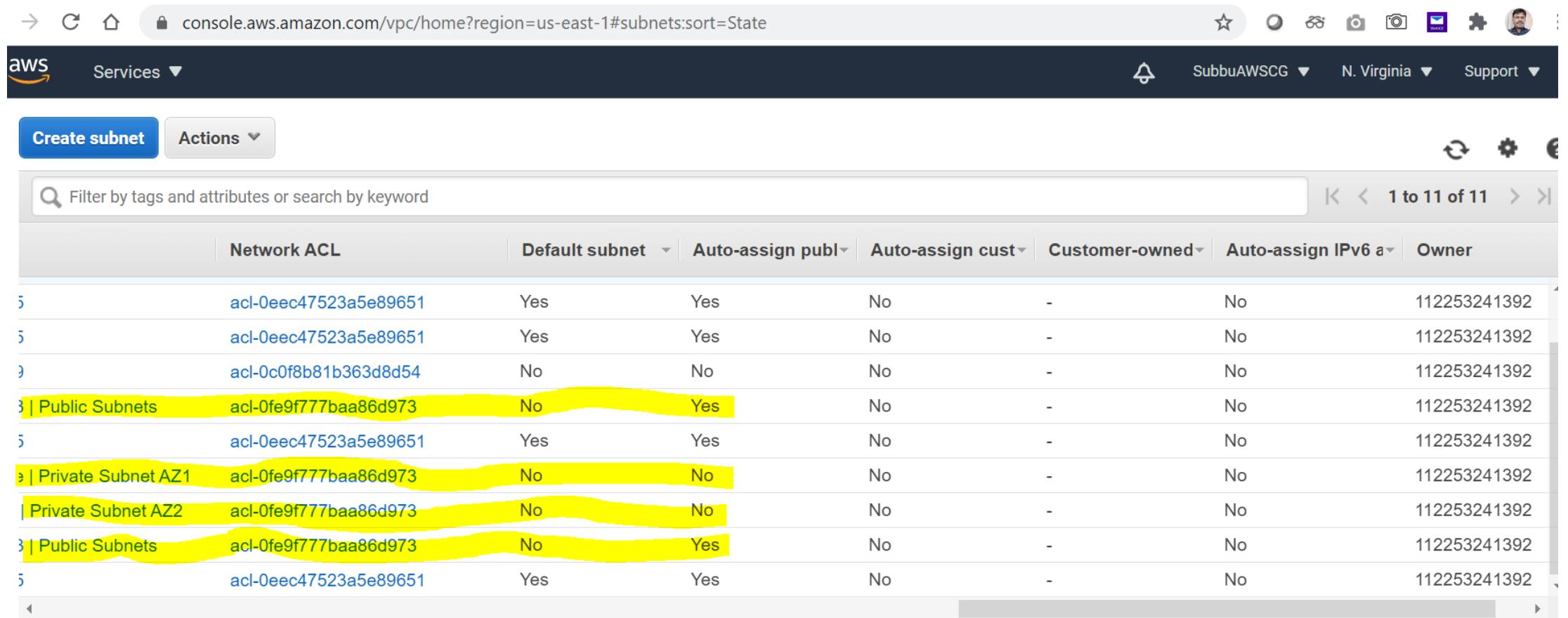
Subnet's and the Route tables:



The screenshot shows the AWS VPC Subnets page in the N. Virginia region. The table lists 11 subnets, each with its ID, CIDR range, availability zone, and associated route table and network ACL. The route table column contains several entries that are highlighted with yellow boxes, specifically 'rtb-092c15e6aa8d40c23' and 'rtb-060da81160ce6a92e'. The network ACL column also contains some highlighted entries.

Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Network Border Group	Route table	Network ACL
4091	-	us-east-1c	use1-az4	us-east-1	rtb-023c615c552198385	acl-0eec47523a5e8
4091	-	us-east-1d	use1-az6	us-east-1	rtb-023c615c552198385	acl-0eec47523a5e8
57	-	us-east-1a	use1-az1	us-east-1	rtb-0a4bc87102b5538c9	acl-0c0f8b81b363d
16374	-	us-east-1a	use1-az1	us-east-1	rtb-092c15e6aa8d40c23 Public Subnets	acl-0fe9f777baa860
4091	-	us-east-1f	use1-az5	us-east-1	rtb-023c615c552198385	acl-0eec47523a5e8
16378	-	us-east-1a	use1-az1	us-east-1	rtb-060da81160ce6a92e Private Subnet AZ1	acl-0fe9f777baa860
16371	-	us-east-1b	use1-az2	us-east-1	rtb-0aa302bf4b6486bcf Private Subnet AZ2	acl-0fe9f777baa860
16377	-	us-east-1b	use1-az2	us-east-1	rtb-092c15e6aa8d40c23 Public Subnets	acl-0fe9f777baa860
4091	-	us-east-1e	use1-az3	us-east-1	rtb-023c615c552198385	acl-0eec47523a5e8

Subnets and their Auto assigned public IP status:



The screenshot shows the AWS VPC Subnets page. The table lists subnets with the following columns: Network ACL, Default subnet, Auto-assign public IP, Auto-assign customer IP, Customer-owned IP, Auto-assign IPv6 address, and Owner. Three specific subnets are highlighted with yellow boxes around their 'Auto-assign public IP' values.

	Network ACL	Default subnet	Auto-assign public IP	Auto-assign customer IP	Customer-owned IP	Auto-assign IPv6 address	Owner
5	acl-0eec47523a5e89651	Yes	Yes	No	-	No	112253241392
5	acl-0eec47523a5e89651	Yes	Yes	No	-	No	112253241392
3	acl-0c0f8b81b363d8d54	No	No	No	-	No	112253241392
3 Public Subnets	acl-0fe9f777baa86d973	No	Yes	No	-	No	112253241392
5	acl-0eec47523a5e89651	Yes	Yes	No	-	No	112253241392
3 Private Subnet AZ1	acl-0fe9f777baa86d973	No	No	No	-	No	112253241392
3 Private Subnet AZ2	acl-0fe9f777baa86d973	No	No	No	-	No	112253241392
3 Public Subnets	acl-0fe9f777baa86d973	No	Yes	No	-	No	112253241392
5	acl-0eec47523a5e89651	Yes	Yes	No	-	No	112253241392

Create Amazon EKS Cluster:

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a 'Services' dropdown, a notification bell, user SubbuAWSCG, region N. Virginia, and support links. The left sidebar features 'Favorites' (Resource Groups & ...), 'Recently visited' (Elastic Kubernetes Service, Console Home, Amazon QLDB, EC2, Secrets Manager, CloudFormation, VPC, Cloud9, IAM, EMR, Data Pipeline, Elasticsearch Service), and a 'Search' bar with the query 'eks'. The main content area displays the 'All services' search results for 'eks', with the 'Elastic Kubernetes Service' result highlighted. Other visible service icons include EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application ..., AWS Outposts, EC2 Image Builder, S3, EFS, FSx, AWS RoboMaker, Customer Enablement, AWS IQ, Support, Managed Services, Blockchain, Amazon Managed Bloc..., Satellite, Ground Station, Amazon SageMaker, Amazon Augmented AI, Amazon CodeGuru, Amazon Comprehend, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Sumerian, Application Integration (Step Functions, Amazon AppFlow, Amazon EventBridge, Amazon MQ, Simple Notification Ser..., Simple Queue Service, SWF), AWS Cost Management (AWS Cost Explorer), and various other services like Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, and AWS Cost Explorer.

Give a name for your EKS cluster:

The screenshot shows the AWS EKS console interface. The top navigation bar includes links for ECS, ECR, and EKS, along with account and region information. On the left, a sidebar for 'Amazon Container Services' lists 'Amazon ECS' (Clusters, Task definitions) and 'Amazon EKS' (Clusters). The main content area features a large title 'Elastic Kubernetes Service (Amazon EKS)' followed by a subtitle 'Fully managed Kubernetes control plane'. Below this, a descriptive paragraph explains that Amazon EKS provides a managed service for using Kubernetes on AWS. A 'Create EKS cluster' form is on the right, with a 'Cluster name' field containing 'SubbuEKSCluster' and a 'Next step' button.

← → ⌂ ⌄ 🔒 console.aws.amazon.com/eks/home?region=us-east-1#/home

aws Services ▾ SubbuAWSCG ▾ N. Virginia ▾

Amazon Container Services X

Containers

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.

Create EKS cluster

Cluster name

SubbuEKSCluster

Next step

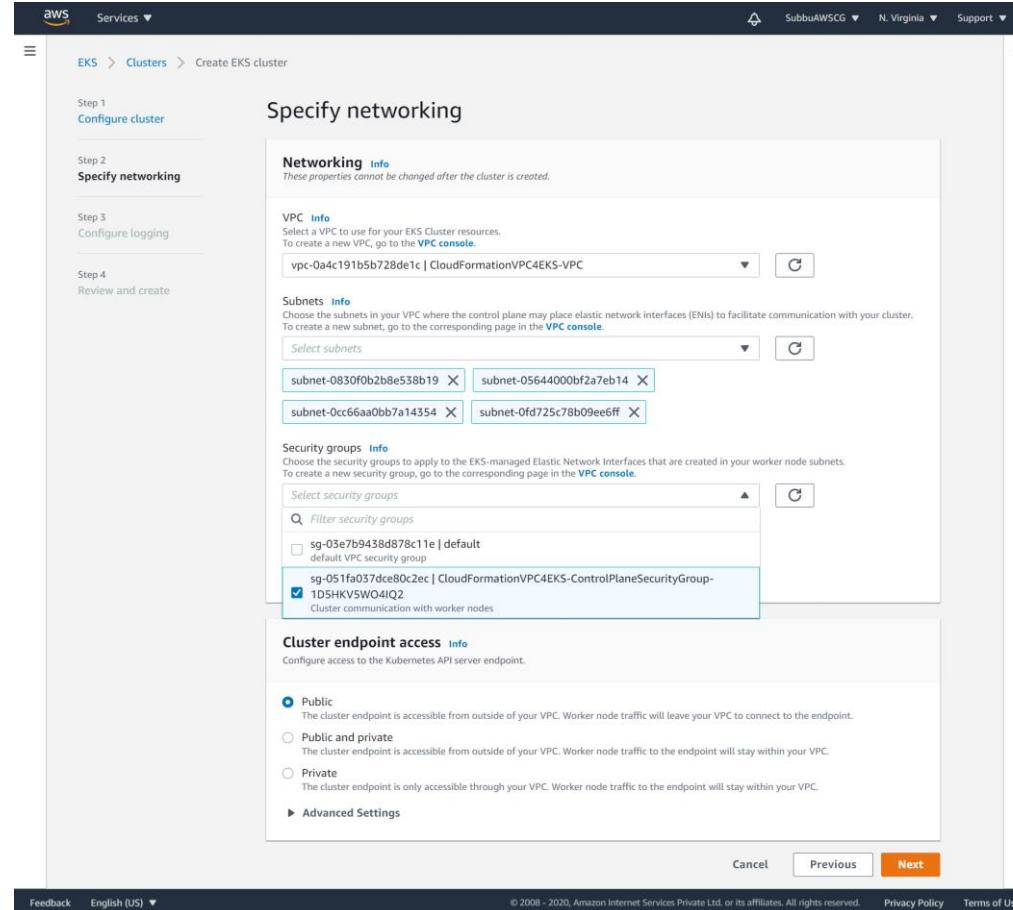
Choose the EKS Role that we created:

The screenshot shows the 'Configure cluster' step of the EKS cluster creation wizard. The left sidebar lists steps: Step 1 (Configure cluster), Step 2 (Specify networking), Step 3 (Configure logging), and Step 4 (Review and create). The main area is titled 'Configure cluster' and contains three sections: 'Cluster configuration', 'Secrets encryption', and 'Tags'. In the 'Cluster configuration' section, the 'Name' field is set to 'SubbuEKSCluster', the 'Kubernetes version' is '1.17', and the 'Cluster Service Role' is 'RoleEKSCluster'. In the 'Secrets encryption' section, there is an option to 'Enable envelope encryption of Kubernetes secrets using KMS'. The 'Tags' section indicates '0' tags and shows a button to 'Add tag'. At the bottom right are 'Cancel' and 'Next' buttons.

Feedback English (US) ▾

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Choose the VPC, Subnet's and Security-Group's that we created:



Configure Log's if needed:

The screenshot shows the AWS EKS 'Create EKS cluster' wizard at Step 1: Configure cluster. The current step is 'Configure logging'. The interface includes a sidebar with navigation links for Step 1 (Configure cluster), Step 2 (Specify networking), Step 3 (Configure logging), and Step 4 (Review and create). The main content area displays the 'Control Plane Logging' configuration section, which allows sending audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs. It includes five log categories: API server, Audit, Authenticator, Controller manager, and Scheduler, each with a 'Disabled' toggle switch.

EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Review and create

Configure logging

Control Plane Logging Info

CloudWatch log group
Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

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

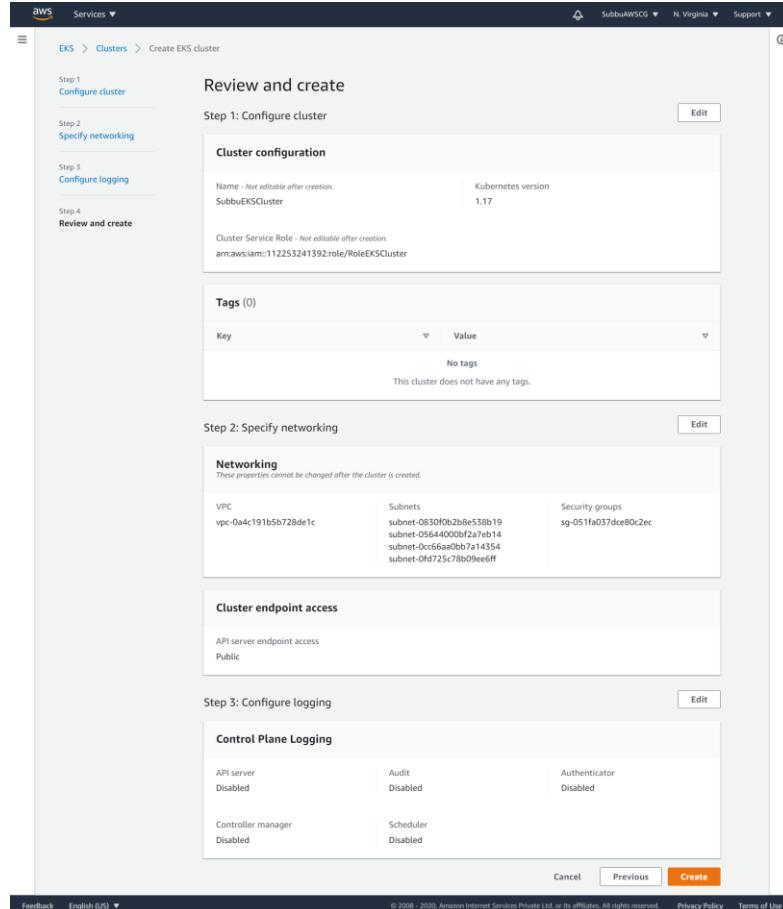
Controller manager
Logs pertaining to state of cluster controllers.
 Disabled

Scheduler
Logs pertaining to scheduling decisions.
 Disabled

Cancel Previous Next

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

Review and Create EKS:



Cluster creation is in Progress:

The screenshot shows the AWS Amazon Container Services (Amazon EKS) Cluster configuration page. At the top, a blue banner displays the message: "Cluster creation in progress" and "SubbuEKSCluster is now being created. This process may take several minutes." Below the banner, the navigation path is shown as EKS > Clusters > SubbuEKSCluster. The main title is "SubbuEKSCluster". On the right, there are "Edit" and "Delete" buttons. The "Cluster configuration" section shows the Kubernetes version (1.17) and Platform version (eks.3), both with "Info" links. The status is listed as "Creating". Below this, a "Details" tab is selected, showing various cluster details in a grid format. The details include:

- API server endpoint
- OpenID Connect provider URL
- Cluster ARN: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
- Creation time: Oct 7th 2020 at 1:18 PM
- Certificate authority
- Cluster IAM Role ARN: arn:aws:iam::112253241392:role/RoleEKSCluster

Now the EKS Cluster is Active - Details:

The screenshot shows the AWS EKS Cluster details page for the 'SubbuEKSCluster'. The cluster is now active, as indicated by the green checkmark icon next to the status 'Active'.

Cluster configuration

- Kubernetes version: 1.17
- Status: Active
- Platform version: eks.3

Details

API server endpoint	OpenID Connect provider URL	Cluster ARN
https://E447BB069F0426F52C2453467FEC385A.gr7.us-east-1.eks.amazonaws.com	https://oidc.eks.us-east-1.amazonaws.com/id/E447BB069F0426F52C2453467FEC385A	arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
Creation time	Certificate authority	Cluster IAM Role ARN
Oct 7th 2020 at 1:18 PM	LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tcK1JSUN5RENDQWJDZOF3SUJBZ0lCQURBTkJna3Foa	arn:aws:iam::112253241392:role/RoleEKSCluster

Navigation and Footer

- Services dropdown: SubbuAWSCG, N. Virginia, Support
- Feedback: English (US)
- Page footer: © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

EKS Cluster – Compute Details:

The screenshot shows the AWS EKS Cluster details page for the cluster "SubbuEKSCluster". The top navigation bar includes the AWS logo, Services dropdown, account name "SubbuAWSCG", region "N. Virginia", and Support link. The main content area displays the cluster configuration and compute details.

Cluster configuration

- Kubernetes version: 1.17
- Status: Active
- Platform version: eks.3

Compute (selected tab)

Node Groups (0)

Group name	Desired size	AMI release version	Launch template	Status
No Node Groups				

This cluster does not have any Node Groups.
Nodes that are not part of an Amazon EKS Managed Node Group are not shown in the AWS console.

Add Node Group

Fargate Profiles (0)

Profile name	Namespaces	Status
No Fargate Profiles		

This cluster does not have any Fargate Profiles.

Add Fargate Profile

Page footer: Feedback, English (US), © 2008–2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved., Privacy Policy, Terms of Use.

EKS Cluster – Networking Details:

The screenshot shows the AWS EKS Cluster Networking Details page for the cluster "SubbuEKSCluster".

Cluster configuration:

- Kubernetes version: 1.17
- Status: Active
- Platform version: eks.3

Networking:

VPC	Subnets	Cluster security group	API server endpoint access
vpc-0a4c191b5b728de1c	subnet-0830f0b2b8e538b19 subnet-05644000bf2a7eb14 subnet-0cc66aa0bb7a14354 subnet-0fd725c78b09ee6ff	sg-0b41525d1f8bdfb74	Public
Service IPv4 range	10.100.0.0/16	Additional security groups	0.0.0.0/0 (open to all traffic)

Networking Options:

- Manage networking

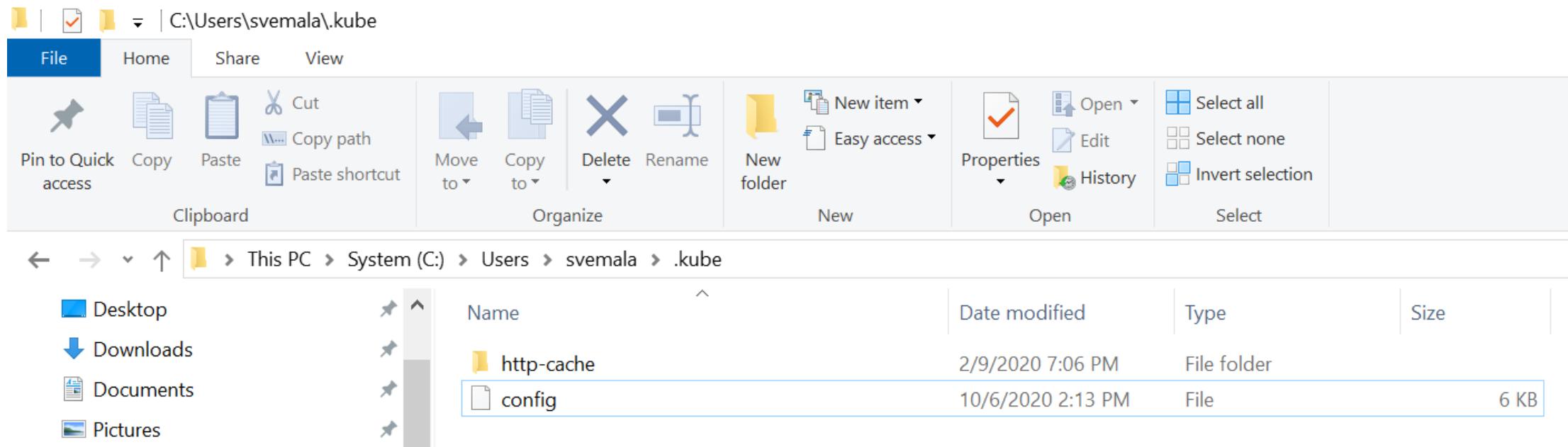
Page Footer:

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

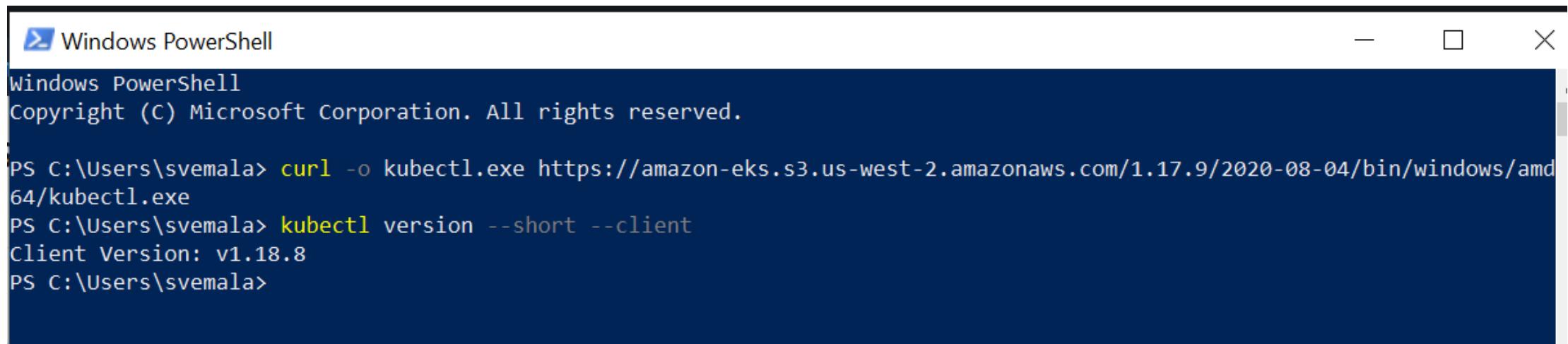
Check for the correct installation of AWS CLI:
Command: aws --version

```
PS C:\Users\svemala> aws --version
aws-cli/1.18.133 Python/3.6.0 windows/10 botocore/1.17.56
PS C:\Users\svemala>
```

Kube config file, after the installation of Kubectl:



Installing Kubectl and check its version:



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "curl -o kubectl.exe https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.9/2020-08-04/bin/windows/amd64/kubectl.exe" being run, followed by "kubectl version --short --client" which returns "Client Version: v1.18.8".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\svemala> curl -o kubectl.exe https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.9/2020-08-04/bin/windows/amd64/kubectl.exe
PS C:\Users\svemala> kubectl version --short --client
Client Version: v1.18.8
PS C:\Users\svemala>
```

Commands and the details:

```
PS C:\Users\svemala> curl -o kubectl.exe https://amazon-eks.s3.us-west-2.amazonaws.com/1.17.9/2020-08-04/bin/windows/amd64/kubectl.exe
```

```
PS C:\Users\svemala> kubectl version --short --client
```

```
Client Version: v1.18.8
```

```
PS C:\Users\svemala>
```

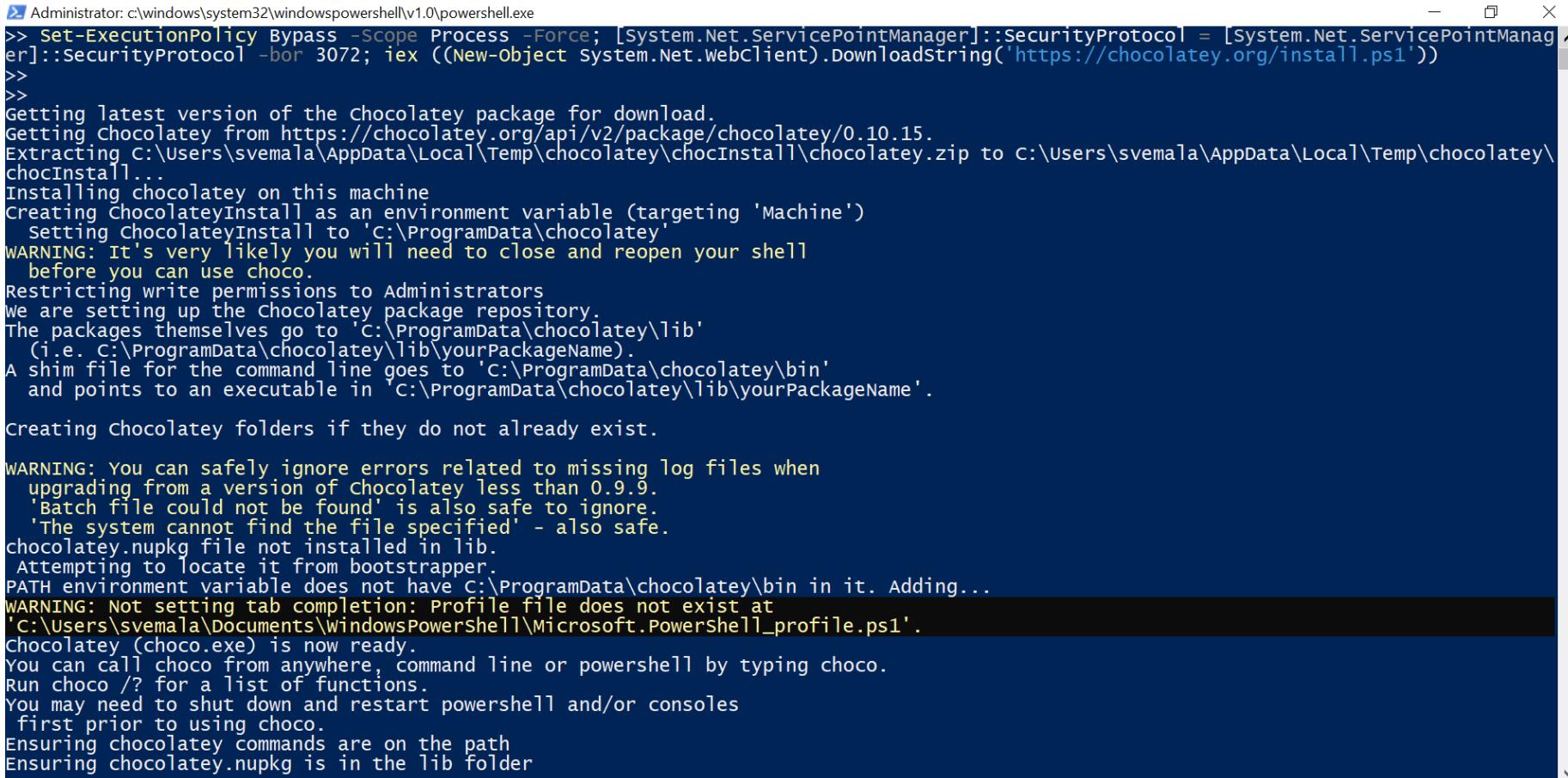
Installing eksctl (Need to be used to configure the EKS through CLI):

Open PowerShell as Administrator and run.

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex
((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Install Chocolatey :

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```



The screenshot shows a Windows PowerShell window with the title bar "Administrator: c:\windows\system32\windowspowershell\v1.0\powershell.exe". The command entered was `Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))`. The output of the script is displayed below:

```
>> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
>>
>>
Getting latest version of the chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.15.
Extracting C:\users\svemala\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\users\svemala\AppData\Local\Temp\chocolatey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
  Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\svemala\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
```

Eksctl is installed:

Command: chocolatey install -y eksctl

Command: eksctl version

```
PS C:\Users\svemala> chocolatey install -y eksctl
Chocolatey v0.10.15
Installing the following packages:
eksctl
By installing you accept licenses for the packages.
Progress: Downloading eksctl 0.29.1... 100%

eksctl v0.29.1 [Approved]
eksctl package files install completed. Performing other installation steps.
eksctl is going to be installed in 'C:\ProgramData\chocolatey\lib\eksctl\tools'
Downloading eksctl 64 bit
  from 'https://github.com/weaveworks/eksctl/releases/download/0.29.1/eksctl_windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\svemala\AppData\Local\Temp\chocolatey\eksctl\0.29.1\eksctl_windows_amd64.zip (19.41 MB)
.
Download of eksctl_windows_amd64.zip (19.41 MB) completed.
Hashes match.
Extracting C:\Users\svemala\AppData\Local\Temp\chocolatey\eksctl\0.29.1\eksctl_windows_amd64.zip to C:\ProgramData\chocolatey\lib\eksctl\tools...
C:\ProgramData\chocolatey\lib\eksctl\tools
  ShimGen has successfully created a shim for eksctl.exe
  The install of eksctl was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\eksctl\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
PS C:\Users\svemala> eksctl version
0.29.1
PS C:\Users\svemala>
```

Installing aws-iam-authenticator: To install aws-iam-authenticator on Windows with Chocolatey.

Command: choco install -y aws-iam-authenticator

```
PS C:\Users\svemala> choco install -y aws-iam-authenticator
Chocolatey v0.10.15
Installing the following packages:
aws-iam-authenticator
By installing you accept licenses for the packages.
Progress: Downloading aws-iam-authenticator 0.5.1... 100%

aws-iam-authenticator v0.5.1 [Approved]
aws-iam-authenticator package files install completed. Performing other installation steps.
Downloading aws-iam-authenticator 64 bit
, from 'https://github.com/kubernetes-sigs/aws-iam-authenticator/releases/download/v0.5.1/aws-iam-authenticator_0.5.1_windows_amd64.exe'
Progress: 100% - Completed download of C:\ProgramData\chocolatey\lib\aws-iam-authenticator\tools\aws-iam-authenticator.exe (38.14 MB).
Download of aws-iam-authenticator.exe (38.14 MB) completed.
Hashes match.
C:\ProgramData\chocolatey\lib\aws-iam-authenticator\tools\aws-iam-authenticator.exe
ShimGen has successfully created a shim for aws-iam-authenticator.exe
The install of aws-iam-authenticator was successful.
Software install location not explicitly set, could be in package or
default install location if installer.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
PS C:\Users\svemala> aws-iam-authenticator help
```

Test the successful installation of aws-iam-authenticator:

Command: aws-iam-authenticator help

```
PS C:\Users\svemala> aws-iam-authenticator help
A tool to authenticate to Kubernetes using AWS IAM credentials

Usage:
  aws-iam-authenticator [command]

Available Commands:
  help      Help about any command
  init      Pre-generate certificate, private key, and kubeconfig files for the server.
  server    Run a webhook validation server suitable that validates tokens using AWS IAM
  token     Authenticate using AWS IAM and get token for Kubernetes
  verify    Verify a token for debugging purpose
  version   Version will output the current build information

Flags:
  -i, --cluster-id ID          Specify the cluster ID, a unique-per-cluster identifier for your aws-iam-authenticator installation.
  -c, --config filename        Load configuration from filename
  --feature-gates map[string]bool : A set of key=value pairs that describe feature gates for alpha/experimental features. options are
                                    AllAlpha=true|false (ALPHA - default=false)
                                    IAMIdentityMappingCRD=true|false (ALPHA - default=false)
  -h, --help                   help for aws-iam-authenticator
  -l, --log-format string     Specify log format to use when logging to stderr [text or json] (default "text")

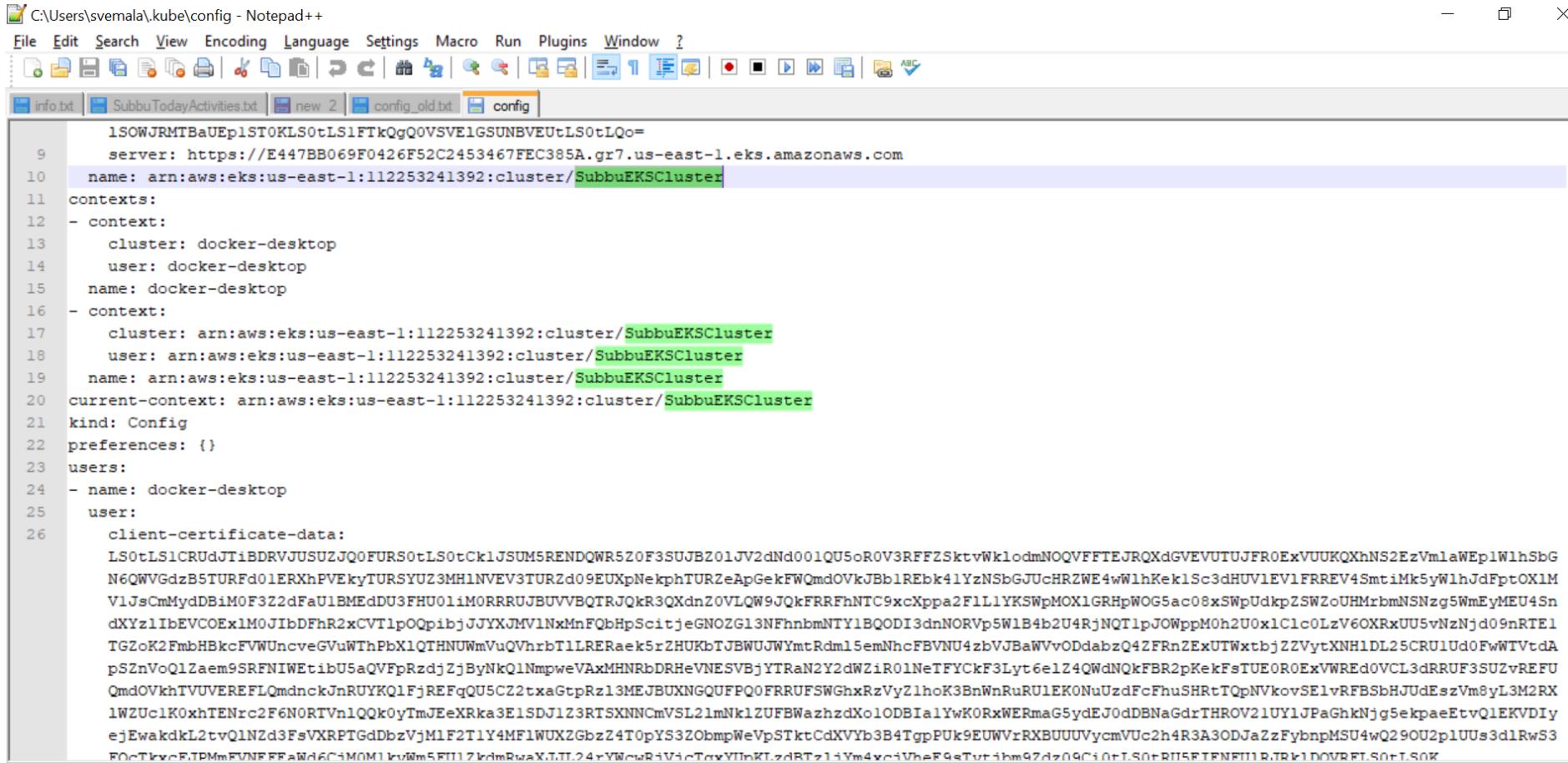
Use "aws-iam-authenticator [command] --help" for more information about a command.
PS C:\Users\svemala>
```

Add created AWS Cluster to the Kubectl config file:

Command: aws eks update-kubeconfig --name SubbuEKSCluster

```
PS C:\Users\svemala> aws eks update-kubeconfig --name SubbuEKSCluster
Added new context arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster to C:\Users\svemala\.kube\config
PS C:\Users\svemala>
```

Kubectl Config file is updated with our cluster:



The screenshot shows a Notepad++ window with the file path C:\Users\svrmala\.kube\config. The file content is a JSON-like configuration for a Kubernetes cluster. The 'SubbuEKSCluster' section is highlighted in green. The configuration includes contexts, users, and preferences. The 'current-context' is set to 'SubbuEKSCluster'. The 'client-certificate-data' field contains a large base64 encoded string.

```
1SOWJRMTBaUEp1ST0KLS0tLS1FTkQgQ0VSVE1GSUNBVEutLS0tLQo=
  server: https://E447BB069F0426F52C2453467FEC385A.gr7.us-east-1.eks.amazonaws.com
  name: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
contexts:
- context:
  cluster: docker-desktop
  user: docker-desktop
  name: docker-desktop
- context:
  cluster: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
  user: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
  name: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
current-context: arn:aws:eks:us-east-1:112253241392:cluster/SubbuEKSCluster
kind: Config
preferences: {}
users:
- name: docker-desktop
  user:
    client-certificate-data:
      LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM5RENQWR5Z0F3SUJBZ01JV2dNd001QU5oR0V3RFFZSkvWklodmNOQVFTEJRQXdGVEVUTUJFR0ExVUUKQXhNs2EzVmLaWEp1WhSbGN6QWVGdzB5TURFd01ERXhPVEkyTURSYUZ3MH1NVEV3TURZd09EUXpNeKphTURZeApGeKFWQmdOVkJBb1REbk41YzNSbGJuHRZWE4wWhKek1Sc3dHUV1EV1FRREV4SmtíMk5yWhJdFptOX1MV1JsCmMydDBiM0F3Z2dFaU1BMEddU3FU01iM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRRfhnTC9xcXppa2F1L1YKSwpMOX1GRHpWOG5ac08xSwpUdkpZSWzoUHMrbmNSNzg5WmEyMEU4SndXYz1IbEVCOEx1M0J1bDFhR2xCVTlpOQpibjJJYXJMViNxMnFQbHpScitjeGNOZG13NFhnbmNTY1BQODI3dnNORVp5W1B4b2U4RjNQT1pJ0WppM0h2U0x1C1c0LzV60XRxUU5vNzNjd09nRTE1TGZoK2FmbHBkcFVWUncveGVuWThPbX1QTHNUWmVuQVhhrbT1LRERaek5rZHUKbTJBWUJWYmtRdm15emNhcFBVNU4zbVJBaWVvODdabzQ42FRnZExUTWxtbjZ2VytXNH1DL25CRU1Ud0FwWTvtdApS2nVoQ1zaem9SRFNIEtibU5aQVfpRzdjZjByNkQ1NmpweVAxMHNrbDRHeVNEsvbJyTRaN2Y2dWZiR01neTFYckF3Lyt6e1Z4QWdNQkfBR2pKekFsTUE0R0ExVWREd0VCL3dRRUF3SUzvREFUQmdOVkhTVUVEREFLQmdnckJnUYKQ1FjREFqQU5CZ2txaGtpRz13MEJBUXNGQUFPQ0FRRUFSWghxRzVyZlhoK3BnWnRuRUEK0NuUzdFcFhuSHRtTQpNVkovSE1vRFBsbHJUdEszVm8yL3M2RX1W2UclK0xhTENrc2F6NORTVn1QQk0yTmJEeXRka3E1SDJ1Z3RTSXNNcmVSL21mNk1ZUFBWazhdXo10DBIa1YwK0RxWERmaG5ydEJ0dBNaGdrTHROV21UY1JPaGhkNjg5ekpaeEtvQ1EKVDIyejEwakdkL2tvQ1N2d3FsVXRPTGdDbzVjM1F2T1Y4MF1WUXZGbZ4T0pYS3ZObmpWeVpSTktCdXVYb3B4TgpPUk9EUWVrRXBUUUUvycmVuc2h4R3A30DJazZzFybnPMsU4wQ290U2p1UUs3d1RwS3FOctTkvxFTPmFVNFFFAWd6C1mOM1kvWm5FU112VdmRwaxTJTT24rVWcwR1ViCtavYInKTzDRTz1iYm4xcivhaf9sTutihm97d709ci0tL50tRm5fTENfII1RTRk1DOVRF1S0tL50k
```

Create Worker nodes:

By using the CloudFormation template from the below S3 location.

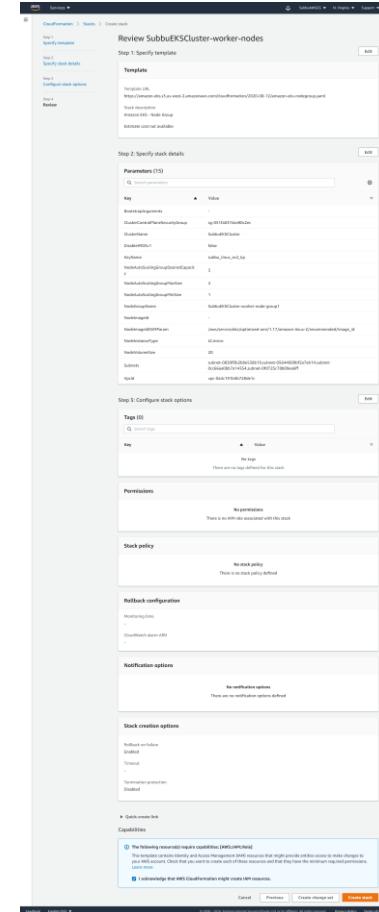
<https://amazon-eks.s3.us-west-2.amazonaws.com/clouformation/2020-08-12/amazon-eks-nodegroup.yaml>

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The left sidebar lists steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main area is titled 'Create stack' and 'Prerequisite - Prepare template'. It explains that every stack is based on a template, which is a JSON or YAML file containing configuration information about AWS resources. Three options are shown: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. Below this is the 'Specify template' section, which asks for a 'Template source'. It shows 'Amazon S3 URL' selected and provides a text input field with the URL: <https://amazon-eks.s3.us-west-2.amazonaws.com/clouformation/2020-08-12/amazon-eks-nodegroup.yaml>. There is also an 'Upload a template file' option and an 'Amazon S3 template URL' input field. At the bottom right of this section is a 'View in Designer' button. Navigation buttons at the bottom are 'Cancel' and 'Next'.

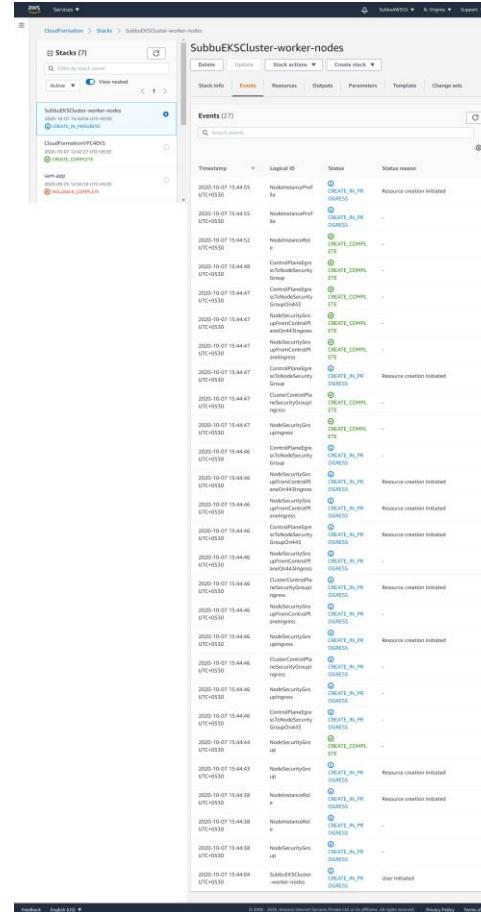
Provide the necessary Auto scaling and other configuration details for the Worker nodes:

The screenshot shows the AWS CloudFormation 'Create stack' wizard at Step 2: Specify stack details. The 'Stack name' field contains 'SubbuEKScluster-worker-nodes'. The 'Cluster name' field contains 'SubbuEKScluster'. The 'Worker Node Configuration' section includes a 'NodeGroupIdentifier' of 'SubbuEKScluster-worker-node-group', a 'NodeAutoScalingGroupMinSize' of '1', and a 'NodeAutoScalingGroupDesiredCapacity' of '3'. The 'NodeAutoScalingGroupMaxSize' is set to '4'. The 'NodeInstanceType' is 't2.micro'. The 'NodeImageIdParam' is '/dev/service/eks/optimized-ami/1.17/amazon-linux-2/recommended/image_id'. The 'NodeImageId' is 'SubbuImageID'. The 'NodeVolumeSize' is '20'. The 'KeyPair' is 'Subbu'. The 'BootstrapArguments' field contains 'Arguments to pass to the bootstrap script. See BootstrapArguments in https://github.com/webscale/eks-ami'. The 'DisableHDSv1' option is set to 'false'. The 'Worker Network Configuration' section includes a 'VpcId' of 'subnet-0350fb0fbef5bbf19 (192.168.0.0/16) (CloudFormation/PC4EKS-VPC)' and a 'Subnets' list containing four subnets: 'subnet-0350fb0fbef5bbf19 (192.168.0.0/16) (CloudFormation/PC4EKS-PublicSubnet01)', 'subnet-05644006fb2fb7e19 (192.168.128.0/16) (CloudFormation/PC4EKS-PrivateSubnet01)', 'subnet-0cc56aa6bb7c74553 (192.168.192.0/16) (CloudFormation/PC4EKS-PrivateSubnet02)', and 'subnet-044725c7fb0be0ef (192.168.64.0/16) (CloudFormation/PC4EKS-PublicSubnet02)'. The 'Next' button is visible at the bottom right.

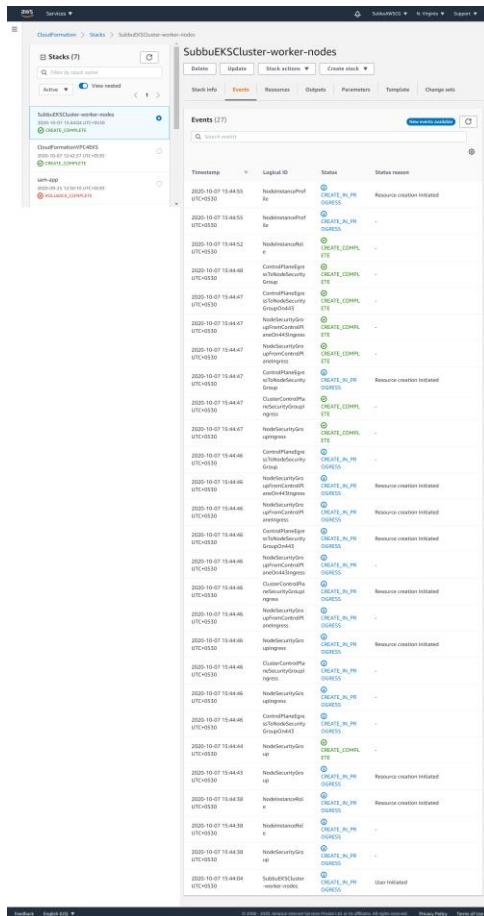
Review the Worker nodes details:



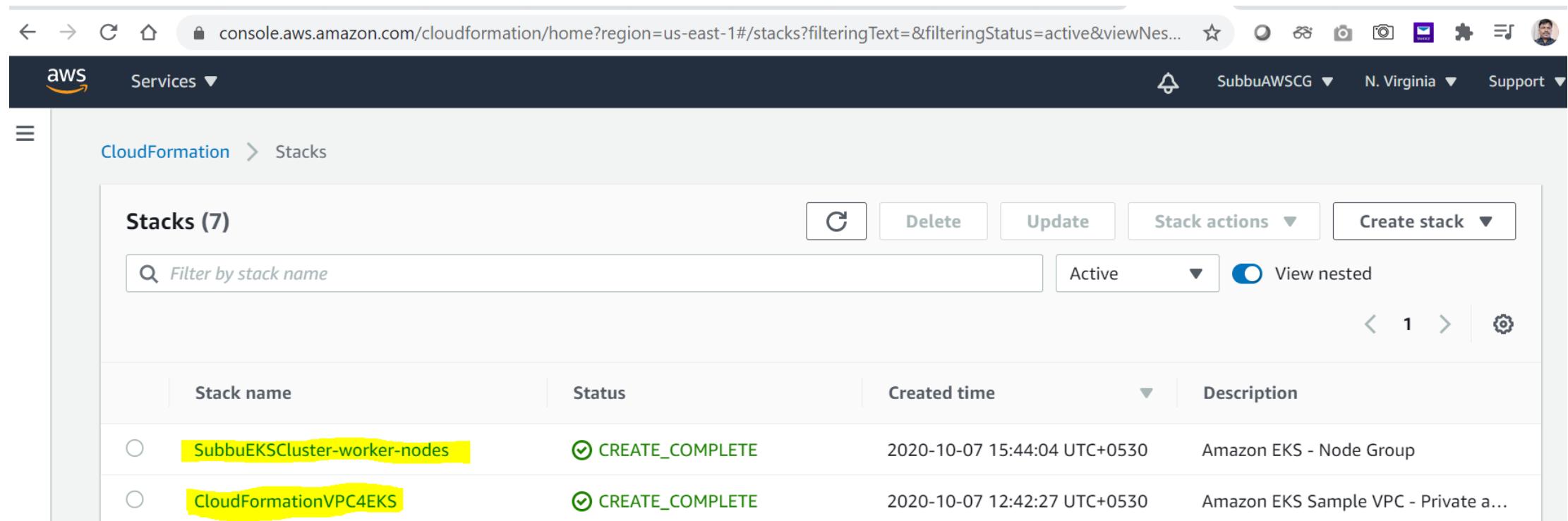
Worker nodes create is in progress:



Worker nodes created:



CloudFormation stacks created for both VPC and for the Worker nodes:



The screenshot shows the AWS CloudFormation Stacks page with the following details:

- Stacks (7)**: The total number of stacks.
- Filter by stack name**: A search bar to filter stacks by name.
- Status**: Active, View nested.
- Stack actions**: Buttons for Delete, Update, Stack actions, and Create stack.
- Table Headers**: Stack name, Status, Created time, Description.
- Stack Details**:
 - SubbuEKSCluster-worker-nodes**: Status: CREATE_COMPLETE, Created time: 2020-10-07 15:44:04 UTC+0530, Description: Amazon EKS - Node Group.
 - CloudFormationVPC4EKS**: Status: CREATE_COMPLETE, Created time: 2020-10-07 12:42:27 UTC+0530, Description: Amazon EKS Sample VPC - Private a...

Check for the created nodes through the Kubectl:

Command: kubectl get nodes

No nodes found, because we need to map the Worker nodes with the EKS Cluster.

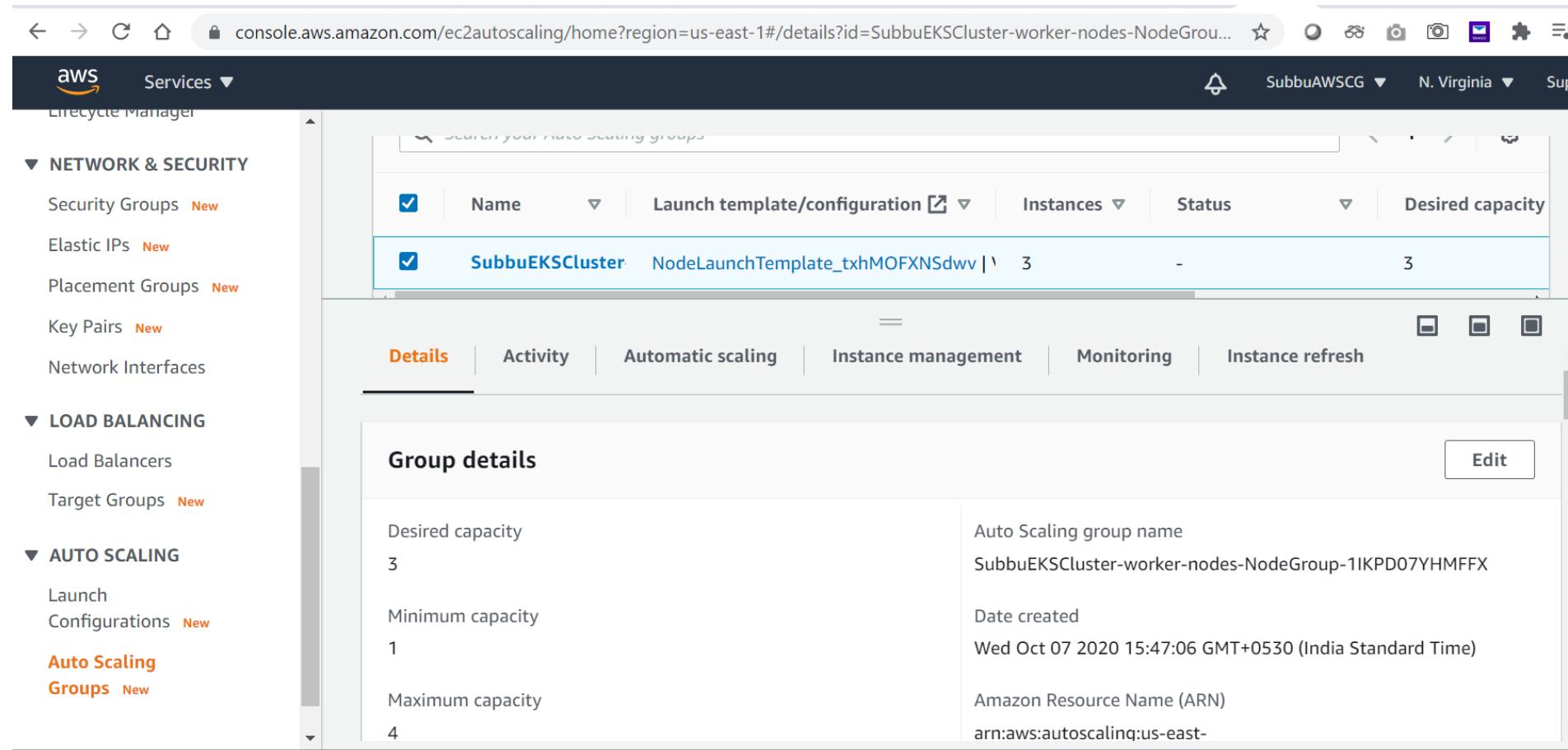
```
PS C:\Users\svemala> kubectl get nodes
No resources found in default namespace.
PS C:\Users\svemala>
```

Worker nodes – ‘Outputs’ tab view:

The screenshot shows the AWS CloudFormation console with the 'Outputs' tab selected for the 'SubbuEKSCluster-worker-nodes' stack. The left sidebar displays a list of stacks, and the main area shows three output entries.

Key	Value	Description	Export name
NodeAutoScalingGroup	SubbuEKSCluster-worker-nodes-NodeGroup-1KPD07YHMFFX	The autoscaling group	-
NodeInstanceRole	arn:aws:iam::112253241392:role/SubbuEKSCluster-worker-nodes-NodeInstanceRole-76NJ9LUILOGR	The node instance role	-

Auto Scaling group created for the Worker nodes:



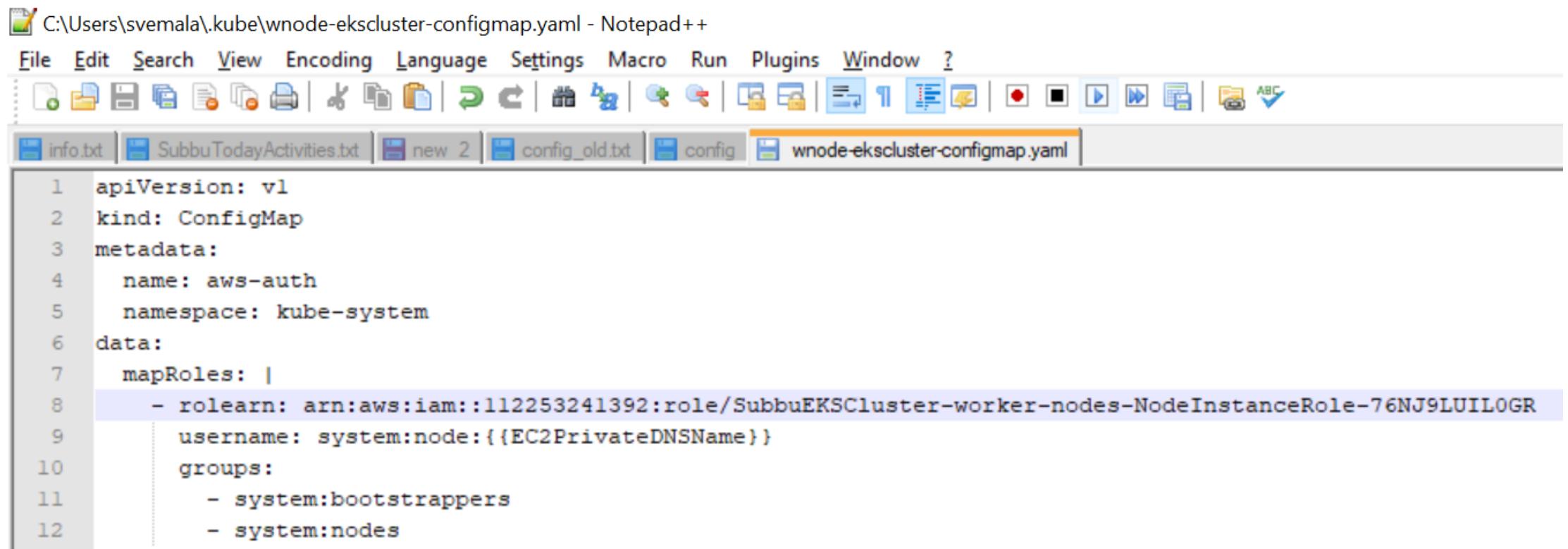
The screenshot shows the AWS Management Console interface for the Auto Scaling service. The left sidebar navigation includes categories like NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. Under AUTO SCALING, 'Auto Scaling Groups' is selected. The main content area displays a table of Auto Scaling groups. One group is selected, showing its details. The group name is 'SubbuEKSCluster'. The 'Launch template/configuration' is 'NodeLaunchTemplate_txhMOFXNSdwv'. It has 3 instances, is in 'In Service' status, and has a desired capacity of 3. Below the table, there are tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The 'Details' tab is active, showing the 'Group details' section with fields for Desired capacity (3), Minimum capacity (1), Maximum capacity (4), Auto Scaling group name (SubbuEKSCluster-worker-nodes-NodeGroup-1IKPD07YHMFFX), Date created (Wed Oct 07 2020 15:47:06 GMT+0530 (India Standard Time)), and Amazon Resource Name (ARN) (arn:aws:autoscaling:us-east-).

Name	Launch template/configuration	Instances	Status	Desired capacity
SubbuEKSCluster	NodeLaunchTemplate_txhMOFXNSdwv	3	In Service	3

Group details

Desired capacity	Auto Scaling group name
3	SubbuEKSCluster-worker-nodes-NodeGroup-1IKPD07YHMFFX
Minimum capacity	Date created
1	Wed Oct 07 2020 15:47:06 GMT+0530 (India Standard Time)
Maximum capacity	Amazon Resource Name (ARN)
4	arn:aws:autoscaling:us-east-

Map Worker nodes with Ekscluster, by giving the ARN of the worker nodes as below in the configmap file:
(Scripts are in my GitHub repository <https://github.com/subbugh/aws-eks>)



The screenshot shows the Notepad++ interface with the file `wnode-ekscluster-configmap.yaml` open. The file contains YAML configuration for a Kubernetes ConfigMap named `aws-auth` in the `kube-system` namespace. The `mapRoles` section defines a role mapping for worker nodes, specifying the IAM role ARN and the node group.

```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: aws-auth
5   namespace: kube-system
6 data:
7   mapRoles: |
8     - rolearn: arn:aws:iam::112253241392:role/SubbuEKSCluster-worker-nodes-NodeInstanceRole-76NJ9LUILOGR
9       username: system:node:{EC2PrivateDNSName}
10      groups:
11        - system:bootstrappers
12        - system:nodes
```

Map the Worker nodes, with the EKS Cluster:

Command: kubectl apply -f wnode-ekscluster-configmap.yaml

Command: kubectl get nodes

```
PS C:\Users\svemala> kubectl get nodes
No resources found in default namespace.
PS C:\Users\svemala> kubectl apply -f wnode-ekscluster-configmap.yaml
error: the path "wnode-ekscluster-configmap.yaml" does not exist
PS C:\Users\svemala> kubectl apply -f wnode-ekscluster-configmap.yaml
configmap/aws-auth created
PS C:\Users\svemala> kubectl get nodes
NAME                      STATUS  ROLES   AGE    VERSION
ip-192-168-237-199.ec2.internal  Ready  <none>  6m34s  v1.17.11-eks-cfdc40
ip-192-168-242-226.ec2.internal  Ready  <none>  6m36s  v1.17.11-eks-cfdc40
ip-192-168-25-115.ec2.internal  Ready  <none>  6m37s  v1.17.11-eks-cfdc40
PS C:\Users\svemala>
```

The EC's created as a part of Worker Nodes as per the Config defined in the yaml file:

The screenshot shows the AWS EC2 Instances page. A blue banner at the top reads: "Welcome to the new instances experience! We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback." Below the banner, the title "Instances (5)" is displayed, followed by a "Filter instances" search bar and a pagination indicator showing page 1 of 1. The main table lists five instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Avail
<input type="checkbox"/>	subbu_linux_...	i-00b93edc00fed1b92	Stopped	t2.micro	-	No alarms +	us-ea
<input type="checkbox"/>	aws-cloud9-S...	i-04577891af8b3cd5	Stopped	t2.micro	-	No alarms +	us-ea
<input type="checkbox"/>	SubbuEKSClu...	i-08bb46f4236bd9a4b	Running	t2.micro	2/2 checks ...	No alarms +	us-ea
<input type="checkbox"/>	SubbuEKSClu...	i-0a4f36c43a2f7ba00	Running	t2.micro	2/2 checks ...	No alarms +	us-ea
<input type="checkbox"/>	SubbuEKSClu...	i-01a8401664511aadd	Running	t2.micro	2/2 checks ...	No alarms +	us-ea

Deploy a sample nginx application:

```
PS C:\Users\svemala> kubectl create namespace my-namespace
namespace/my-namespace created
PS C:\Users\svemala>
```

Create a sample application: Use the sample yaml deployment file from the github.

```
PS C:\Users\svemala> kubectl create namespace my-namespace
namespace/my-namespace created
PS C:\Users\svemala> kubectl apply -f aws-eks-sample.yaml
service/my-service created
deployment.apps/my-deployment created
PS C:\Users\svemala>
```

Check for the POD's created:
Command: kubectl get all -n my-namespace

```
PS C:\Users\svemala> kubectl get all -n my-namespace
NAME                                         READY   STATUS    RESTARTS   AGE
pod/my-deployment-65dcf7d447-6b5xh         1/1     Running   0          2m12s
pod/my-deployment-65dcf7d447-r6x87         1/1     Running   0          2m12s
pod/my-deployment-65dcf7d447-r9bm5         1/1     Running   0          2m12s

NAME           TYPE      CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
service/my-service ClusterIP  10.100.208.68 <none>        80/TCP   2m13s

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-deployment  3/3     3            3           2m13s

NAME                           DESIRED  CURRENT   READY   AGE
replicaset.apps/my-deployment-65dcf7d447  3        3         3        2m13s
PS C:\Users\svemala>
```

View the details of the deployed service:

Command: kubectl -n my-namespace describe service my-service

```
PS C:\Users\svemala> kubectl -n my-namespace describe service my-service
Name:                  my-service
Namespace:             my-namespace
Labels:                app=my-app
Annotations:           selector: app=my-app
Type:                  ClusterIP
IP:                    10.100.208.68
Port:                  <unset>  80/TCP
TargetPort:            80/TCP
Endpoints:             192.168.229.142:80,192.168.230.108:80,192.168.28.128:80
Session Affinity:      None
Events:                <none>
PS C:\Users\svemala>
```

View the details of one of the pods that was deployed:

Command: kubectl -n my-namespace describe pod my-deployment-65dcf7d447-6b5xh

```
Administrator: c:\windows\system32\windowspowershell\v1.0\powershell.exe
PS C:\Users\svemala> kubectl -n my-namespace describe pod my-deployment-65dcf7d447-6b5xh
Name:           my-deployment-65dcf7d447-6b5xh
Namespace:      my-namespace
Priority:      0
Node:          ip-192-168-237-199.ec2.internal/192.168.237.199
Start Time:    Wed, 07 Oct 2020 16:59:36 +0530
Labels:         app=my-app
                pod-template-hash=65dcf7d447
Annotations:   kubernetes.io/psp: eks.privileged
Status:        Running
IP:            192.168.229.142
IPs:
  IP:          192.168.229.142
Controlled By: Replicaset/my-deployment-65dcf7d447
Containers:
  nginx:
    Container ID:  docker://9638d6778308ce1215bd874db7eea080b80e5edc81369ef101c18de1d446b820
    Image:         nginx:1.19.2
    Image ID:     docker-pullable://nginx@sha256:c628b67d21744fce822d22fdcc0389f6bd763daac23a6b77147d0712ea7102d0
    Port:          80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Wed, 07 Oct 2020 16:59:42 +0530
    Ready:         True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-277v5 (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready      True
  ContainersReady  True
  PodsScheduled  True
Volumes:
  default-token-277v5:
    Type:      Secret (a volume populated by a secret)
    SecretName: default-token-277v5
```

Continued - More details of the POD 'my-deployment-65dcf7d447-6b5xh':

```
Administrator: c:\windows\system32\windowspowershell\v1.0\powershell.exe
nginx:
  Container ID: docker://9638d6778308ce1215bd874db7eea080b80e5edc81369ef101c18de1d446b820
  Image: nginx:1.19.2
  Image ID: docker-pullable://nginx@sha256:c628b67d21744fce822d22fdcc0389f6bd763daac23a6b77147d0712ea7102d0
  Port: 80/TCP
  Host Port: 0/TCP
  State: Running
    Started: Wed, 07 Oct 2020 16:59:42 +0530
  Ready: True
  Restart Count: 0
  Environment: <none>
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from default-token-277v5 (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  default-token-277v5:
    Type:     Secret (a volume populated by a Secret)
    SecretName: default-token-277v5
    Optional:  false
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute for 300s
              node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type      Reason     Age   From                               Message
  ----      -----     --   ----
  Normal    Scheduled  13m   default-scheduler                Successfully assigned my-namespace/my-deployment-65dcf7d447-6b5xh
  to ip-192-168-237-199.ec2.internal
  Normal    Pulling    13m   kubelet, ip-192-168-237-199.ec2.internal  Pulling image "nginx:1.19.2"
  Normal    Pulled    13m   kubelet, ip-192-168-237-199.ec2.internal  Successfully pulled image "nginx:1.19.2"
  Normal    Created    13m   kubelet, ip-192-168-237-199.ec2.internal  Created container nginx
  Normal    Started    13m   kubelet, ip-192-168-237-199.ec2.internal  Started container nginx
PS C:\Users\svemala>
```

Enter into any of the POD's:

Command: `kubectl exec -it my-deployment-65dcf7d447-6b5xh -n my-namespace -- /bin/bash`

```
PS C:\Users\svemala> kubectl exec -it my-deployment-65dcf7d447-6b5xh -n my-namespace -- /bin/bash
root@my-deployment-65dcf7d447-6b5xh:/#
```

View the DNS resolver configuration file:
Command: cat /etc/resolv.conf

```
PS C:\Users\svemala> kubectl exec -it my-deployment-65dcf7d447-6b5xh -n my-namespace -- /bin/bash
root@my-deployment-65dcf7d447-6b5xh:/# cat /etc/resolv.conf
nameserver 10.100.0.10
search my-namespace.svc.cluster.local svc.cluster.local cluster.local ec2.internal
options ndots:5
root@my-deployment-65dcf7d447-6b5xh:/#
```

Remove the sample service, deployment, pods, and namespace:

Command: kubectl delete namespace my-namespace

```
PS C:\Users\svemala> kubectl delete namespace my-namespace
namespace "my-namespace" deleted
PS C:\Users\svemala> kubectl get services -o wide
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE      SELECTOR
kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP      5h16m   <none>
nginx     ClusterIP  10.100.165.129  <none>        80/TCP       17m     run=nginx
PS C:\Users\svemala> kubectl -n my-namespace describe pod my-deployment-65dcf7d447-6b5xh
Error from server (NotFound): namespaces "my-namespace" not found
PS C:\Users\svemala> kubectl get nodes
NAME                  STATUS  ROLES      AGE      VERSION
ip-192-168-237-199.ec2.internal  Ready   <none>    163m    v1.17.11-eks-cfdc40
ip-192-168-242-226.ec2.internal  Ready   <none>    163m    v1.17.11-eks-cfdc40
ip-192-168-25-115.ec2.internal  Ready   <none>    163m    v1.17.11-eks-cfdc40
PS C:\Users\svemala> kubectl get all -n my-namespace
No resources found in my-namespace namespace.
PS C:\Users\svemala>
```

Delete the Cluster view on the AWS console:

The screenshot shows the AWS EKS Clusters page. A prominent blue banner at the top states "Cluster termination in progress" and "SubbuEKSCluster is now being terminated. This process may take several minutes." On the left, a sidebar menu for "Amazon Container Services" includes "Amazon ECS", "Clusters", and "Task definitions". Under "Amazon EKS", "Clusters" is selected and highlighted in orange. The main content area displays a table titled "Clusters (1)" with one item: "SubbuEKSCluster" (Cluster name), "1.17" (Kubernetes version), and "Deleting" (Status). There are "Info", "Edit", "Delete", and "Create cluster" buttons above the table. A search bar is also present.

Cluster name	Kubernetes version	Status
SubbuEKSCluster	1.17	Deleting

Delete Auto Scaling group for the Worker nodes:

The screenshot shows the AWS EC2 Auto Scaling Groups page. At the top, there is a navigation bar with the URL `console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/details?id=SubbuEKSCluster-worker-nodes-NodeGrou...`. Below the navigation bar is the AWS logo and a "Services" dropdown menu. A modal window titled "New EC2 Experience" is open, asking "Tell us what you think".

The main content area displays "Auto Scaling groups (1/1)". There is a search bar with the placeholder "Search your Auto Scaling groups". Below the search bar is a table header with columns: Name, Launch template/configuration, Instances, Status, and Desired capacity. The table contains one row for the "SubbuEKSCluster" group, which has a Launch template/configuration of "NodeLaunchTemplate_txhMOFXNSdwv" and 3 instances.

Below the Auto Scaling groups section is another table titled "Instances (3)". This table lists three instances with the following details:

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configuration
i-00a6fbef9...	InService	t2.micro	-	NodeLaunchTemplate_txhMOFXNSdwv

Release the EIP's created along with VPC:

The screenshot shows the AWS EC2 console interface for managing Elastic IP addresses. The URL in the browser is `console.aws.amazon.com/ec2/v2/home?region=us-east-1#Addresses:`. The top navigation bar includes links for Services, SubbuAWSCG, N. Virginia, and Support. On the left, a sidebar under the Instances heading lists various options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts. The main content area is titled "Elastic IP addresses (2/2)". It features a search bar labeled "Filter Elastic IP addresses". Below the search bar is a table with columns: Name, Allocated IPv4 add..., and Type. Two rows are listed: one with IP `3.220.233.208` and another with IP `54.161.31.126`, both marked as Public IP. To the right of the table is an "Actions" menu with options: View details, Release Elastic IP addresses (which is highlighted with a yellow box), Associate Elastic IP address, and Disassociate Elastic IP address. The "Release Elastic IP addresses" option is the target of the user's action.

Name	Allocated IPv4 add...	Type
-	3.220.233.208	Public IP
-	54.161.31.126	Public IP

Actions

- View details
- Release Elastic IP addresses**
- Associate Elastic IP address
- Disassociate Elastic IP address

EIP's released:

The screenshot shows the AWS EC2 console interface. The left sidebar has sections for Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing. The main content area has a green banner at the top stating "Elastic IP addresses released." followed by two IP addresses: 54.161.31.126 and 3.220.233.208. Below the banner is a table titled "Elastic IP addresses" with columns: Name, Allocated IPv4 add..., Type, and Allocation ID. A search bar and pagination controls are also present.

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Addresses:

aws Services SubbuAWSCG N. Virginia Support

Snapshots Lifecycle Manager

Network & Security

- Security Groups New
- Elastic IPs** New
- Placement Groups New
- Key Pairs New
- Network Interfaces

Load Balancing

Elastic IP addresses released.
Elastic IP addresses 54.161.31.126, 3.220.233.208

Name	Allocated IPv4 add...	Type	Allocation ID
			No E

Refer to my GitHub repo, for the script files and other details:

<https://github.com/subbugh/aws-eks>

Appendix: URL's and some more Slides, for more information.

<https://github.com/subbugh/aws-eks>

<https://docs.aws.amazon.com/eks/index.html>

<https://aws.amazon.com/getting-started/hands-on/deploy-kubernetes-app-amazon-eks/>

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

<https://chocolatey.org/install>

<https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html>

<https://docs.aws.amazon.com/eks/latest/userguide/create-public-private-vpc.html>

<https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html>

<https://docs.aws.amazon.com/eks/latest/userguide/launch-workers.html>

<https://docs.aws.amazon.com/eks/latest/userguide/sample-deployment.html>

<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

<https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/>

<http://docs.aws.amazon.com/autoscaling/latest/userguide/as-process-shutdown.html>

<https://docs.aws.amazon.com/eks/latest/userguide/delete-cluster.html>

<https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/#scaling-the-application-by-increasing-the-replica-count>