Journal of Mathematical & Computer Applications

Review Article



Open d Access

Automating The Deployment of MERN Stack on AWS App Runner Using AWS Code Pipeline

Aauti

USA

ABSTRACT

This paper explores the deployment of web applications utilizing AWS App Runner, a managed platform that simplifies the deployment process by allowing developers to select runtimes and deploy applications without the need for extensive configuration. It focuses on the advantages of using Docker runtime for API execution and web application deployment, highlighting the ease of dockerizing applications and the seamless integration with Amazon Elastic Container Registry (ECR). ECR provides a fully-managed Docker container registry, facilitating the efficient storage, management, and deployment of Docker container images. Additionally, the paper discusses the role of AWS CodeCommit as a secure, scalable, managed source control service for hosting private Git repositories, emphasizing its significance in the deployment pipeline. The combined use of these AWS services offers a streamlined, cost-effective solution for deploying scalable and secure web applications directly from source code or container images to the AWS Cloud, thereby enhancing the deployment process for developers.

*Corresponding author

Aauti, USA.

Received: January 17, 2024; Accepted: January 23, 2024, Published: January 30, 2024

Keywords: MERN, AWS, Cloud Computing, Programming, Software Development

In this paper, we explore the adoption and implementation of AWS App Runner, a managed platform service by Amazon Web Services, tailored for streamlined application deployment. AWS App Runner stands out for its capability to simplify the deployment process by allowing developers to choose the desired runtime environment, such as NodeJS, or leverage Docker to containerize and deploy applications, particularly focusing on the MERN stack. This service eliminates the complexities typically associated with configuration and management, thereby facilitating the deployment of APIs and web applications directly from source code or container images. Additionally, it ensures scalability and security within the AWS Cloud ecosystem.

The utility of AWS App Runner extends to its seamless integration with Docker, where Docker images, including those from Amazon's Elastic Container Registry (ECR), can be effortlessly deployed. Our study delineates a comprehensive automation strategy encompassing the entire deployment lifecycle of a MERN Stack application using Docker within the AWS App Runner environment. This encompasses leveraging AWS CloudFormation for resource provisioning and adopting continuous integration and continuous deployment (CI/CD) methodologies to achieve efficient and automated application deployment. This paper aims to provide a detailed exploration of the end-to-end automation process, highlighting the practical and theoretical aspects of deploying a containerized MERN Stack on AWS App Runner, thereby demonstrating the efficiency and efficacy of managed platform services in modern cloud environments.

- Prerequisites
- Example Project
- Setup a MongoDB Atlas
- Build For Production
- Externalize Environment Variables
- Dockerize the project
- Running WebApp on Docker
- Creating ECR with CloudFormation
- Pushing Docker Image to ECR
- Deploy CF Template through CLI
- Setup CodeCommit Repos
- Create a ServiceRole for CodePipeline
- Setup CI/CD With AWS CodePipeline for ECR Deployment
- Setup CI/CD With AWS CodePipeline For AppRunner WebApp
- Testing the WebApp
- Summary
- Conclusion

Prerequisites

For individuals embarking on their journey in web development, it is recommended to consult the following resource for a comprehensive guide on developing and constructing applications using the MERN stack. This material serves as an essential foundation for beginners in the field.

• How To Develop and Build MERN Stack

(https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-mern-stack-9a7a1099624)

Docker Essentials

You need to understand Docker concepts such as creating images, container management, etc. Below are some of the links that you can understand about Docker if you are new.

- Docker Docs
- (https://docs.docker.com/)
- Docker A Beginners Guide
- (https://medium.com/bb-tutorials-and-thoughts/dockera-beginners-guide-to-dockerfile-with-a-sample-project-6c1ac1f17490)
- Docker Image Creation and Management (https://medium.com/bb-tutorials-and-thoughts/dockerimage-creation-and-management-9d91e4c277b1)
- Docker Container Management (https://medium.com/bb-tutorials-and-thoughts/dockercontainer-management-with-examples-c280906158a8)
- Understanding Docker Volumes (https://medium.com/bb-tutorials-and-thoughts/ understanding-docker-volumes-with-an-exampled898cb5e40d7)

AWS Prerequisites

Amazon Web Services (AWS), recognized as a pioneer in the cloud computing domain, offers an extensive portfolio of over 200 services. It is crucial for users to understand and select the appropriate AWS services that align with their specific requirements. If you are new to AWS or just getting started you can see the following article.

• How To Get Started with AWS

(https://medium.com/bb-tutorials-and-thoughts/how-to-get-started-with-aws-9731a4f855a7)

Example Project

Here is an example of a simple tasks application that creates, retrieves, edits, and deletes tasks. We actually run the API on the NodeJS server, and you can use MongoDB to save all these tasks.

	MERN	Stack Exa	ample		
ToDation A					
IODO LIST					
Greate a Task		Insignee			
tons					
To the Devra					
Survey .					
Tasks					
Tauk isl	Task Kame	Amigree	Status		
610912266e6ce34672740304	9639339	(1)(1)	III Progress	108	DOM:N

MERN Stack Example

Here is a GitHub link to this project. You can clone it and run it on your machine.

// clone the project
git clone https://github.com/bbachi/mern-stack-example
// React Code
cd ui
npm install
nom start
// API code
cd ani
nom install
npm mada
npm run dev

Setup a MongoDB Atlas

The core of MongoDB Cloud is MongoDB Atlas, a fully managed cloud database for modern applications. Atlas is the best way to run MongoDB, the leading modern database. There are two ways to deploy MongoDB on AWS and you can check them here on this page. We are using a fully-managed MongoDB Cluster for this post. Let's create your MongoDB Account here. You can either log in with any of your Gmail accounts or you can provide any other email address to create the account.

mongoDB.	MongoDB Stands with the Black Community	*
G Log is with Georgie	Join MongoDB in supporting organizations that are fighting for racial justice and equal opportunity.	
Drail Address ()	Jain Now +	
70mt Don't have an account? Sign Up		

MongoDB Login

Once you log in with your account you will see the dashboard below where you can create clusters. Let's create a cluster called todo-cluster by clicking on build a cluster and selecting all the details below. Make sure you select AWS Cloud.

Serveriess	Dedicated	Shared
Fer learning and exploring MongeDB No credit card required to start. Upgn Explore with sample datasets. Limit of	in a sandbox environment. Basic configur ade to dedicated clusters for full functions f one free cluster per project.	ation controls. slity.
Cloud Provider & Region		AWS, N. Virginia (us-east-1)
aws or o	ogle Cloud	
* Recommended region 🕘 🙁 Dedi	icated tier region 🕘 🖉 Carbon emission	n doto unavailable ()
N. Virginia (us-cost-1) #	■ Paris (cu-wost-3) ★	Sydney (ap-southeast-2) *
Tregen (us-west-2) #	Frankfurt (eu-central-i)	ASIA
Chie Lus-cest-2) *	Frankfurt (eu-central-1)	🕼 Hong Kong (op-east-1) 🖈
Cregon (us west-2) Crego	Frankfurt (eu-central-i) Steckholm (eu-north-i) *	ASUA ■ Heng Kang (op-east-1) ★ Tekye (op-northeast-1) ★
Cregon (ut west-2) Crego	Frenkfurt (su-central-1) Stockhalm (su-north-1) * I li leland (su-west-1) * Stockhalm (su-west-2) * 23	ASIA Hong Kang (op-east-1) + Tekye (op-northeast-1) + Kumbel (op-south-1) +
Cregan (in west 2) Company (in west 2) Company (in west 2) Company (in the second of the secon	Frenkfurt (zu-centrol-1) Frenkfurt (zu-cen	ALLA I Hang Kang (sp-east.)) * Takye (sp-northeast.)) * Hembel (sp-south.i) * (st) Secul (sp-northeast.2) *

MongoDB Pricing

Make sure you select the Cloud Environment since we are deploying this on AWS Cloud. You can click on the connect button to see the details about connecting to the cluster. You need to create a user and Allow Access from anywhere for now.

rs one - more on	Setup connection security	Choose a connection meth	od Connect			
stanister.	You need to secure your MongoDB cluster now. Read more 2	Atlas cluster before you can	use it. Set which us	ers and IP addresses can ac	cess your	
odo-cluster	You can't connect yet. Set up yo	ur firewall access and user s	ecurity permission t	selow.		
100 9-2-11	Add a connection IP add	ess				
MARCH MEANING	Add Your Current IP Address	Add a Different IP Add	ress Allow Act	cess from Anywhere		
STER TIER Sandbor (Gemera MON	Create a Database User					
13 / N. Vegene (un E Sica Set - 3 podas	This first user will have all Keep your credentials handy	asAdmin (3 ^e permissions fo you'll need them for the new	r this project. step.		Enhance alled throughput options. up	You riche grade
ED REALM APP	Username	Pass	word			Upp
	admin		4	SHOW		
				Create Databar	e User	
					_	
	Close			Choose a connectio	n method	

Connect to Cluster

You can see three ways of connecting to the cluster on the next screen. We have created a cluster and it's time to create a database. Click on the collections to create a new database as below. I have given a database name as tasks and the collection name is todos.



Creating a Database

Let's insert the first document into the collection by clicking the button insert document. We have seen three ways we can connect to this cluster and read the collections. Let's connect to the database with Mongo Compass.

todo-cluster					4.2.11	AW
Overview Real Time	Metrics Collections P	vofiler Performance	Advisor Onl	ine Archive	Command Line Tools	
TABASES: 1 COLLECTIONS: 1					Let V	ISUALIZI
+ Create Database	tasks.todos	DOCUMENTS 0 INDEXES FOR	N. 5175 4KB			
taska todos	Find Indexes	Schema Anti-Patterns 🔘	Appregation	Search Indexes		
	QUERY RESULTS 1-1 OF 1	imple")				
	_id:0bject2d["3fe4m mane:"nomel" description:"hame 1 createdus:020-06- createdus:"tirst im opdatedus:202-06- spdatedby:"first im	000307acc578520ac4") 0cscription" 0cr6100100.000-00100 tr" 00705100100.000-00100 tr"				

Creating a Collection

The first thing we need to do is to download and install Mongo Compass from this link. Let's get a connection string from the Atlas dashboard as below.

	Setup connection security Choose a connection method Connect		
Q Feet a statter	Choose a connection method Vew documentation (2)		
SANDBOX	Get your pre-formatted connection string by selecting your tool below.		
• todo-cluster	Connect with the mongo shell Interact with your cluster using MongoDB's interactive Javancript interface	>	
CLUSTER THER MO Standbox (Gamers	Connect your application Connect your application to your cluster using MongoDB's native drivers	>	
REGION ANNS / N. Wignus fus TYPE Replics Set - 3 model	Connect using MongoDB Compass Explore, modify, and visualize your data with MongoDB's QUI	>	Enhance Your E and throughput, richer m options, sograde po
Nime Linked	Go Back	Close	Upped
Disked Means are Nires Liviant	Go Back	Close	Uppe

Connection Ways

You can see the same collection in the MongoDB Compass as well. Here is the connection string that you can connect to MongoDB.

mongodb+srv://admin123:admin123@todo-cluster.zpikr. mongodb.net/?retryWrites=true&w=majority

Local	talks todos Documenta =	2
(OFRICETE)	tasks.todos	000046455 1 1678 1678 HOLESS 1 20.0KB 20.0KB
Q. Filter your data	Documents Aggregations Schema Explain	Plan Indexes Validation
> admin > config		+ OPTIONS
> local	ADD DATA · A VEW III 0 II	Displaying documents 1 - 1 of 1 C > Q REFRESH
Todos	Late (Supervise) descriptions (There: Exercic) (Supervise) created area; 2024-81-81 (Supervise), etc., etc41, etc. created area; 2024-81-81 (Supervise), etc., etc41, etc. created area; 2024-81-81 (Supervise), etc., etc41, etc. supervise), "Supervise", etc., etc., etc., etc41, etc. supervise), "Supervise", etc., etc	

Mongo Compass UI

Build for Production

Numerous approaches exist for constructing a MERN Stack for production deployment, with the optimal strategy varying based on the specific use case or deployment environment. This paper delineates various methodologies for preparing the MERN Stack for production use.

• How to Build MERN Stack for Production

(https://medium.com/bb-tutorials-and-thoughts/how-to-build-mern-stack-for-production-1462e70a35cb)

Externalize Environment Variables

Reading environment variables is one of the most common things that we do when we are building apps. It doesn't matter whether you are developing front end app or back-end API you have so many variables that should be outside of your application source code that makes your app or API more configurable. For example, if you want to hide logger statements in production or do something else based on the environment you can pass this as an environment variable. If you want to change later all you need to change is in one place.

• Reading Environment Variables in NodeJS api

(https://medium.com/bb-tutorials-and-thoughts/readingenvironment-variables-in-nodejs-rest-api-e75bb04b813d)

When it comes to this application, there are two environment variables, one is the Mongo Connection string, and another one is PORT.

PORT=80

MONGO_CONNECTION_STRING=mongodb+srv://admin123:admin123<u>@todo</u>cluster.zpikr.mongodb.net/?retryWrites=true&w=majority

You must put these in the webpack.config.js file so that these values are used when we dockerize the app for production.

• Webpack.config.js file

(https://gist.github.com/bbachi/aa25aec5b82320d28cc5ee137bb 8b8cf#file-webpack-config-js)

Dockerize the Webapp

Amazon EKS is a managed service that makes it easy for you to run Kubernetes on AWS. The first thing you need to do is to dockerize your project.

We use multi-stage builds for efficient docker images. Building efficient Docker images are very important for faster downloads and lesser surface attacks. In this multi-stage build, building a React app and putting those static assets in the build folder is the first step. The second step involves building the API. Finally, the third step involves taking those static build files and API build and serving the React static files through the API server.

We need to update the server.js file in the NodeJS API to let Express know about the React static assets and send the index. html as a default route. Here is the updated server.js file. Notice the line numbers 41 and 20.

• Server.js file

(https://gist.github.com/bbachi/e828985a85cfb9da08164afa885 49211#file-server-js)

Let's build an image with the Dockerfile. Here are the things we need for building an image.

In constructing a production-ready application using the MERN Stack, the process begins with a base image of node:14-slim. Initially, both package.json files—one for the Node.js server and the other for the React UI-are copied into the Docker file system, with dependencies installed to enhance build speed for subsequent changes. This preemptive step prevents the redundancy of reinstalling dependencies with each source modification. Following this, all source files are copied, and dependencies installed, culminating in the execution of npm run build to generate the React application assets within a 'build' folder inside the 'ui' directory. The second stage also utilizes the node: 14-slim base image, focusing on the Node.js environment by copying its package.json into an './api' directory, installing necessary dependencies, and incorporating the server is file into this directory. The final stage combines the elements, starting again with the node:14-slim image, to amalgamate the built UI and API files, concluding with the command node api.bundle.js to run the bundled server application, thereby streamlining the deployment of the MERN Stack for production.

Here is the Complete Dockerfile link where you can run on your machine.

• Docker file

(https://gist.github.com/bbachi/06eecfc6c956d01c99180523c26

77c15#file-dockerfile)

Let's build the image with the following command.

// build the image docker build -t mern-image .

// check the images docker images

Running the Webapp on Docker

Once the Docker image is built. You can run the image with the following command.

// run the container docker run -d -p 80:80 --name mern-stack mern-image

// list the container docker ps

// logs docker logs mern-stack

// exec into running container
docker exec -it mern-stack /bin/sh

You can access the application on the web at this address $\mbox{http://localhost}$

	MEDN				
	WERN	Stack Exa	mpie		
ToDo List					
Task		Assignee			
Create a Task		Assignee			
Status:					
To Be Done					
Submit					
Trate					
lasks					
Task Id	Task Name	Assignee	Status		
	asdasd	acdacdead	To Be Done	 and the second	

Example Project

Creating ECR with Cloud Formation

First, you need to understand the anatomy of the CloudFormation template. We can't go through everything here you can look at the AWS Cloudformation docs here.

AWSTemplateFormatVersion: "version date" Description: String Metadata: Template Metadata Parameters: Set of Parameters Rules: Set of Rules Mappings: Set of Mappings Conditions: Set of Conditions Transform: Set of Transforms Resources: Set of Resources Outputs: Set of Outputs

The only required one is the Resources of all these options in the template file. Below is the template YAML file with which we are creating the ECR repository through CloudFormation. The first one is the version and description. The version has only one value and in the description, you can put anything about

your repo or deployment. Since it's an ECR Repository, I have given the following description. The next main thing is the Resources section. Since we are creating only one resource which is AWS AppRunner, I have added one resource called ECRRepo. You can name it anything you want and the type is obviously AWS::ECR::Repository. The main thing here is adding a policy text where the users specified only can push the image into the repository. In production, you should create a role here.

• ecr-template.yaml

(https://gist.github.com/bbachi/f47addffb396297502fd8789b75d 7cdc#file-template-ecr-yaml)

The output section contains the ARN of the ECR repository. Let's create this stack through AWS Console. You can do it either console or AWS CLI. You can click on the Create Stack button. On the next screen, you must upload the above YAML file by selecting the second option.

Specify template	Create stack						
Step 2 Specify stack details	Prerequisite - Prepare template	Prerequisite - Prepare template					
Step 1 Configure stack options	Prepare template Every stack is based on a template. A template is a	JSDN or 10045. Na that contains configuration inform	ation about the AMS resources you want to include in the stack.				
Ship 4 Review	Template is ready	O We a sample template	Create template in Designer				
	Specify template A template is a XON or XAN. No that describes yo	ur stack's manures and properties.					
	Template source Selecting a template generates an Amazon 53 URL	where it will be stored.					
	 Amazon S3 URL 	O Uploa	d a template file				
	Upload a template file Choose file (a) SION or XXM, forwarded file	r.					
	STURP Interview and Transmission	un li f. templates, 1 jun Nivennii, un east. 2720	22067EX-template-ecryami View in Designer				

Creating a Stack

Let's give a stack name on the next screen.

Specify template	Specify stack details
Stop 2 Specify stack details	Stack name
	Stack name
Confinues stack nations	ecr-docker-apprunner
1001000000	Stack name can include letters (A-2 and a z), numbers (0-3), and dashes (-).
Step 4 Review	Parameters
	Parameters are defined in your template and allow you to input outcom values when you create or update a stack.
	No parameters
	There are no parameters defined in your template

Stack Details

Configure the Stack options on the below screen.

Conditionation () Stack () Online task Stard () Stack () Online task Stard () Stack () Configure stack options Stard () St

Configure the Stack

You can see the output ARN in the outputs section.

Stacks (1)	Stack info Even	s Resources Outputs	Parameters Template	Change sets	Delete
Active • O Viewnested) Outputs (1)				
eor-docker-apprunner 2022-03-07 21 48:14 UTC-0600	Q Search outputs				
© CREATE_COMPLETE	Key A	Value		w.	Description
	ECRRepolym	amawsecrus-east-2.86422792	19192 repository/vest-api		ServiceArm of ECR 8

Output ARN

Let's go and check the ECR console to see if this repository is created or not.

es, feotures	s, blogs, docs, and more	(Option+5)				0 4 1	9 on •	admin1 @ 8642-2792
Amazon	CCR > Repositories							
Privat	Public							
Beller	ate reportionier (1	1	C	View	with commends	materia de la	nine w	Create monthage
Q	Cost repairments		•		Them continues		10	< 1 > 6
	Repository name 🔺	un	Created at	v	Tag immutability	Scan frequency	Encryption type	Pull through cache
	webapp	(3) 864227929192.dkr.ecr.us-east- 2.amazonaws.com/webapp	September 03, 2022, 10:13:58 (UTC-07)		Enabled	Scan on push	AES-256	Inactive

ECR Console

Pushing Docker Image to ECR

We have created an ECR repository in the above section. Let's create a docker image from the example project section above with the following command.

docker build -t webapp.

You can view further instructions after creating the Docker image in the top right corner.

feeture	s, blogs, docs, and more	(Option+5)				• •	O Onio •	admin1 @ 8642-271
kenazon	ECR > Repositories							
Prive	nte Public							
Priv	vate repositories (1)	of 1)	C	View	push commands	Delete A	ctions 🔻 🔽	reate repository
Q,	Find reprisiteries			1				<1>
	Repository name A	ulu	Created at	v	Tag immutability	Scan frequency	Encryption type	Pull through cache
0	webapp	864227929192.dki.ecc.us-east- 2.amazonaws.com/webapp	September 03, 2022, 10:13:58 (UTC-07)		Enabled	Scan on push	AES-256	Inactive

Viewing Push Commands

You should authenticate first, then tag and finally push the docker image. Let's follow these commands.



Push Commands

You can tag and push the image with the following command.

docker tag webapp:latest 864227929192.dkr.ecr.us-east-2.amazonaws.com/webapp:v1

docker push 864227929192.dkr.ecr.us-east-2.amazonaws.com/webapp:v1

Once the image is pushed, you can view it on the ECR Console.

woh:	200							View	such commands Ed
veba	app								
Imag	ges (1)							C	Duleta Scan
Q,	Find images								$\langle 1 \rangle$
	Image tag	Artifact type	Pushed at	•	Size (HB)	 Image URI	Digest	Scan status	Vulnerabilities
	*1	Image	September 05, 2022, 10:42:10 (utfc-07)		66.50	C ^{Copy}	🗇 sha256:56233ccla214ad9	Complete	A S High = 65 others (detaild)

Image Pushed

Creating CloudFormation Template

We need to create multiple resources for the AppRunner Template, which we will go through one by one in this section. The initial ones are the version, description, and parameters. The version has only one value and in the description, you can put anything about your repo or deployment. Since it's a NodeJS REST API on App runner, I have given the following description. You can provide the parameters while deploying the template.

Container Image: This is the ECR Image URL. You can fetch it from the ECR Console.

Environment: The environment you want to deploy this stack such as dev, test, prod, etc.

Welcome Message: This is the environment variable you want to pass while deploying the template.

Image Repository Type: There are two types that the App Runner service accepts at this time of writing: ECR and ECR_PUBLIC

You can have conditions on your template so that we can execute something based on that. You can define that under the section called Conditions. Since the App Runner only accepts ECR and ECR_PUBLIC we are putting a condition for that. Here is a complete YAML file.

• ECR Template YAML

(https://gist.github.com/bbachi/9cf4e6a88ca6414426f869b47d5 3ac34#file-template-yaml)

AWS Cloud Formation Command's

Here are some of the commands that you can run through AWS CLI to create and update the stack. You can explore more on AWS Docs.

	// create-stack aws cloudformation create-stack stack-name myteststack template-body file:///home/testuser/mytemplate.json parameters ParameterKey=Parm1,ParameterValue=test1 ParameterKey=Parm2,ParameterValue=test2
	// listing stacks aws cloudformation list-stacks
	// describing stack aws cloudformation describe-stack-eventsstack-name <stack name=""></stack>
	// updating stack aws cloudformation update-stackstack-name <i>mystack</i> template-url <>
	// validating template aws cloudformation validate-template
]	Deploy CF Template Through CLI

Let's create a resource through CLI with the following command. Make sure you update the command with your path of the file location.

aws cloudformation create-stack

- --stack-name nodejs-restapi
- --template-body <file://file-location>

--parameters ParameterKey=Environment,ParameterValue= dev ParameterKey=WelcomeMessage,ParameterValue="Welcome from the CLI"

You can see the resources created on the respective screens below.

🗇 Stacks (2)	C	nodejs-restapi	Outrus Parameters	Investore Channes with	Delete Update Stack actions * Cr
Active • O View rested	(1)	Events (11)			
nodeju-vistapi 2002-05-08.2211518-070-0400	0	Q. Seech ments			
O CHEATE_EDMPLETE		Tinestang v	Legical ID	Status	Status reason
ect-docker-apprunner 2022-03-07-211.46.14 UTC-0608		2022-03-08 22 18:23 UTC-0600	redejs-redags	O-CHEATE,COMPLETE	•
CHENT COMPLETE		2022-03-08 22:18:19 UTC-0600	Node/SilestAM	@CHEATE_COMPLETE	
		2022-03-08 22:14:28 UTC-0600	Node:SRestRM	OCHEATE IN PROCAESS	Amounte creation initiated
		2022-05-06 22:14:17 UTC-0600	Node/Silvestil/1	OCHLATE, N. PROGAESS	8
		2022-03-00 22:14:16 UTC-0600	Accessitale	@CHARLCOMPLETE	<u>×</u>
		2022-03-08 22-14 16 UTC-0600	Instancellale	O CHLATE, COMPLETE	
		2022-03-08 22:14:02 UTC-0600	InstanceRole	@CHEATE, IN, PRODUCTS	Resource creation initiated
		2022-03-08 22:14:02 UTC-0600	Accessifiale	O CHEATE IN PROCAESS	Resource creation initiated
		2032-03-08 22:14:01 UTC-0600	InstanceRola	@CHEATE_IN_PROCHESS	*
		2022-03-08 22:14:01 UTC-0600	AccessRole	CREATE_IN_PROCRESS	*
		2022-05-08 22:11 58 UTC-0600	nodejs-restapi	O CHEATT IN PROCHESS	Quer Initiated

Cloud Formation Events

You can see the outputs listed on the AWS CloudFormation console.

🖸 Stacks (2)	3	webapp			Delete	Update	•	Stack actions
Q. Filter by stack norme		Stack info Events Resources Outputs Parameters	Template	Change set	8			
Active • View nested	5	Outputs (5) Q. Search outputs						
webapp 2022-09-03 2010-21 UTC-0700 @ CREATE_COMPLETE	•	Key A Value AppRamerServiceAr amawcappnimerus-east-28642279291923	ervice/WebApp-		v Des	cription	×	Export nam
ecr-docker-apprunter 2022-09-03-10:15:53-070-0700		n dev/612ac55de876452088320a724e65abdf AppRumerServiceld 612ac55de876452088320a724e65abdf			App	riceAm of Gi RunnerServi	tHub Iceld	-
O CREATE_COMPLETE		AppRunnerServiceUni dilepidiett2 us exist-2 awsapprunner.com			400	RunnerServi	inthis	

Template Created

	MERN S	stack Exa	imple	
ToDo List				
Task		Assignee		
Create a Task		Assignee		
Status:				
To Be Done				
Submit				
and a local second s				

Project Running Through AWS App Runner

Setup CodeCommit Repos

The first step of CI/CD and automation flow is setting up the repositories on the AWS CodeCommit. AWS CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories.



Code Commit Process

We are going to create two repositories: one for MERN Stack and another one for Cloudformation templates. Make sure you have Administrator access or access to the Codecommit to create the repos for the AWS user you created. I have AdministratorAccess that allows me to create repositories.

carrier j	1		
	User ARN Path Creation time	amawsiam:864227923192user/admin1 @ / 2021-03-06 65:48 PDT	
Permissions	Groups (1) Tags	Security credentials Access Advisor	
- Permissi	ions policies (3 policie	s applied)	
Add permis	sions		
Policy	name •		Policy type +
Attached fro	m group		
• 🔒 Am	azonEC2ContainerRegistryF	TufAccess	AWS managed policy from
+ 😝 Adr	ninistratorAccess		AWS managed policy from
+ 🤨 Am	azonEC2ContainerServiceto	vEC2Role	AWS managed policy from
Barminai	ons boundary (not set		

Permissions

You can go to the AWS CodeCommit and create a repository below.

-	Neveloper Tools) CodeCommit) Repositories) Create repository				
1	Create repository				
ļ	inste a carura reportions to store and share your orde. Basis by tuning a reportions prove	a descelation	for your		
1	epository. Repository names are included in the URLs for that repository.	a sesciption	roi your		
	Repository settings				
	Repository name				
	300 characters maximum. Other limits are be				
	Description - optional				
	1,000 characters maximum				
	Tags				
	Add				
	Enable Amazon CodeGuru Reviewer for Java and Python - optional				
	Get recommendations to improve the quality of the Java and Python code for all pull				
	requests in this repository.				
	A service-linked role will be created in IAM on your behalf if it does not exist.				
				_	

Creating a Repository

Once created, you can clone it in different ways using HTTPS, SSH, and HTTPS (grc). You can use any of these methods to connect to this repository. If you want to use HTTPS, you must create Git credentials for your IAM user in the IAM section. Finally, I have created two repositories, and we can push the code later once we go through other important sections. You can copy all the source code from the example project in the above section to the AWS CodeCommit repository below.

app_	_webapp	⇔ Netify ▼	master	•	Create pull request	Clone URL ¥
app.	_webapp trise					Add file 🔻
	Name					
•	api					
	ui					
0	DS_Store					
٥	gitignore					
D	docker-compose.yaml					
D	Dockerfile					
D	README.md					
REA	DME.md				View sour	ce Edit

Let's create another repository called app_cf_templates and push these two Yaml files into the repo. We can put these templates in this repo.

• Template-ecr.yaml

(https://gist.github.com/bbachi/0d983547fd99f056ea21b880e73 013f6#file-template-ecr-yaml)

• Teamplet-apprunner.yamlfile (https://gist.github.com/bbachi/3e bbee23c3c1fefb2fb7319f9f5f21a2#file-template-apprunner-yaml)

Create a Service Role for Code Pipeline

We have pushed the code to the AWS CodeCommit Repos in the above section. It's time to create a service role for the AWS Codepipeline to create the required resources when you run the pipeline.

Let's go to IAM dashboard Click on Roles and create role. Add these policies.

AmazonEC2ContainerRegistryFullAccess AWSAppRunnerFullAccess AWSCloudFormationFullAccess AWSAppRunnerServicePolicyForECRAccess AWSCodeCommitFullAccess AmazonS3FullAccess

AWSCodePipelineServiceRole-us-east-1-app-ec	r-deployment	Delet
Summary		Ed
Shaton das Dastor 22, 1022, 13.19 (J/10-07.00) Janf Jachniy § 1 Than aga	ARN (2) amaxwalam.3642221921182.role/ser Maximum session duration 1 Neur	nce-role/XW3Code/hpelineService/hole-us-east-1-app-eor-deplaymen
Permissions Trust relationships Tags Access Advisor: Revoke sess	sions	
You can attach up to 10 managed policies.		Sensate Remove Add permissions •
Q. Filter policies by property or policy name and press enter.		< 10 1
Q. Filter policies by preparity or policy name and press write. Policy name (2		< 1 > 0 9 Description
Q., Film policies by property or policy name and press write: Policy name (2 @ MADDoalCouncil Ulteress	T Type AND managed	 t > 0 ** Description Provides full access to ARS Code/Commit via the.
O, File polices by property or policy mans and press write Policy name 27 O A ANDCOdeConnerf-MAccess O A ANDCOdeConnerf-MAccess O A ANDCOdeConnerf-MAccess	17 Type AKS managed AKS managed	 1 > 0 ** Description Provides ML access to ANS Cool/Commit via the Provides administrative access to Anson CDI r.
Cr. Prinz policies by property large trains and prinze	 Type Add managed Add managed Add managed Add managed 	 t -> 0 ** Description Provides M access to AMS Cold/Commt via the Provides daministrative access to AMS Cold/Commt via the Provides Malaccess to all locality via the AMS M.
Char politica by properly ur polity nerve and years where Polity years (2 Or Add/out-charved/uldcenes Or Annual Collicitation Markenes Or Annual Collicitation Markenes Or Annual Collicitation Or Annual Collicitation	 Type AKS managed AKS managed AKS managed AKS managed 	Peorlysian Provides Laf access to ASIS ColorCorrent on the Provides administration access to ASIS ColorCorrent on the Provides administration of the ASIS M Geneta germissions to all Access to a buckness to all Access to a buckness
Cher polities by properly set parent and press write Policy parent (2 Diff. Add/Calor Annual Advances Diff. Add/Calor Advances Diff. Add/Calor Annual Advances Diff. Add/Calor Advances Diff. Advances Diff. Advan	▼ Type ANS managed ANS managed ANS managed ANS managed ANS managed	Description Provides Auf access to AVIS DesCentral via the Provides administrative access to Avianne CORF. Provides Auf access to AVIS DescRemention.
C. Files parties for property or pointy news and/press and/or Painty parties (2) E E E Anticolocity of property full Access E E B E Anticolocity of property full Access	 Type Add-manged Add-manged Add-manged Add-manged Add-manged Add-manged Add-manged Add-manged 	Description Provide la descess to ANS Conflictment eters Provides to ANS P

Service Role

You need to create one custom inline policy for AWS Codepipeline to create necessary roles while running the AWS CloudFormation. I have called this policy iam-create-role



Setup CI/CD with AWS CodePipeline for ECR Deployment We have pushed the code to the AWS CodeCommit Repos in the above section. Access the CodePipeline dashboard below and click on the button Create pipeline.



You can name the pipeline anything and you can select the servicerole created above or you can choose to create a new service role and the above policies to the role.

P Pipeline settings Pipeline settings Control the plottine same after it is created. Control the plottine same. The cancel will be plottine same after it is created. Control the plottine same. The cancel will be plottine same after it is created. Control to plottine same after it is created. Cont	
dd Source stage Pipetine name Date the pupeter name. Date the name then 15 the stagethere name after 11 is created. Date the name then 15 the name then 15 the name then Date the name then 15 the name the na	
dd build stop: apg_ers_deplayment_pipeline top 4 Tromm that 100 Australs dd daplay tog 4 Service role top 5 O Fuisting service role O Fuisting service role	
tag 4 Te server than 100 characters	
Add deploy stage Service role	
tas 5 O New service role O Existing service role	
Create a service role in your account Choose an existing service role from your account	
Role ARN	
Q, am:aws:lam:864227929192.role/service-role/AWSCodePipelineServiceRole-ur ×	
Advanced settings	

Pipeline Settings

You can click on the next button and choose the source where you read the code. I have selected AWS CodeCommit.

Choose pipeline settings	Add source stage Mo	
Step 2 Add source stage	Source	
Step 3 Add build stage	Source provider This is where you started your input artifacts for your pipeline. Choose the provider and then provide the	conection details.
Step 4	9	
Add deploy stage	AWS CodeCommit	
Step S Review	Amazon ECR	Previous Next
	Amazon S3	
	Bitbucket	
	GitHub (Version 1)	
	GitHub (Version 2)	
	GitHub Enterprise Server	

Selecting Source

Select the right repository and the branch you want to deploy.

Step 1 Choose pipeline settings	Add source stage տ					
Step 2 Add source stage	Source					
Step 3 Add hulld states	Source provider This is where you stored your input artifacts for your pipeline. Choose	the provider and then provide the connection details.				
read and the property of	AWS CodeCommit	•				
step 4 Add deploy stage	Repository name Choose a repository that you have already created where you have pushed your source code.					
Step 5	Q, app_cf_templates	×				
never w	Branch name Choose a branch of the repository					
	Q. master	×				
	Change detection options Choose a detection mode to automatically start your pipeline when a	change occurs in the source code.				
	 Amazon CloudWatch Events (recommended) Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs 	AWS CodePipeline Use AWS CodePipeline to check periodically far changes				
	Output artifact format Choose the output artifact format.					
	 CodePipetine default MVS CodePipetine uses the default zip format for artifacts in the pipetine. Does not include git metadata about the repeation; 	Full clone AWK Code/Peeline passes metadata about the repository that allows subsequent actions to do a full git clone. Only supported for AWS Codebuild actions.				

Source Stage

Since we are running the AWS Code Formation to create the ECR, we can skip the build stage.

Choose pipeline settings	Add build stage 📷
Step 2 Add source stage	Build - optional
Step 3 Add build stage	Build provider This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names:
Step 4	· · · · · · · · · · · · · · · · · · ·
Add deploy stage	
Step S	Cancel Previous Skip build stage Next
No. of Contract of	

Skip Build Stage

Skip build stage	×
Your pipeline will not include a build stage. Are you	sure you want to skip this stage?
	Cancel Skip

Skip Build Stage

You can't skip the deployment stage. I have chosen AWS CloudFormation as a Deployment provider. Make sure you select the right template from the repository.

Choose pipeline settings	Add deploy stage 📷
Step 2 Add source stage Step 3	You cannot skip this stage Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.
Add build støge Step 4 Add depløy stage	Deploy
Step 5	Deploy provider Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
Review	AWS CloudFormation
	Region
	US East (N. Virginia)
	Action mode When you update an existing stack, the update is permanent. When you use a change set, the result provides a diff of the updated stack an the original stack before you choose to execute the change.
	Create or update a stack
	Stack name If you are updating an existing stack, choose the stack name.
	Q ecr-deployment X
	Template Socioly the template you uploaded to your source location. Artifact name File name Template file path
	SourceArtifact v template-ecxyaml SourceArtifact.template
	Temptase configuration is eastload Specify the embyoarism file you splaced to your source lecation.
	Artifact name File name Template configuration file

Choose the right role that we created above.

	v	Template configuration file path	
Capabilities - optional Specify whether you want	to allow AWS CloudFormation to	o create IAM resources on your behalf.	
		*	
CAPABILITY_IAM X			
Role name			
Q arn:aws:iam::864	227929192:role/service-role	e/AWSCodePipelineServiceRole-u: 🗙	
Output file name			
Output file name	on		
Output file name File generated by this actic Advanced	on		

More Settings

Once you confirm everything and create a pipeline, you can see the pipeline created successfully and in progress status.

nelsper Teols X odePipeline	Social Congratulational The pipeline app_arc_doployment_pipeline has been coasted.	Create a santituation role for this pipelin
Seures + CodeCammit Artifiants + CodeArtifact	DeveloperTexts > Contractions > Peptides > app_ecr_deptionent_pipeline	😞 Worlfy 🔻 🛛 Edit 🔰 Step execution 🔹 Glasse pipeline 🔹 Relater Charge
Build + CodeDuild Beplay + CodeDuplay	O Source apagent	
Populine + CodePopuline Getting started Populates Populates Minters	Search @ And Colorman O Ben Han To annucleur pri	
Settings Settings	Dudie taestiye	
Coltoreniume Freedback	O Deplay mennun	
	Despiny () Pet Conditionation () Of Date Nam No watching pet	

Pipeline in Progress

If everything is successful	l, you	can s	ee the	pipeline	successful
below.					

eveloper Tools) CodeRipeline) Pipelines) app_ecr_deployment_pipeline	
app_ecr_deployment_pipeline	A Notify + Edit Step centurien Clone pipeline Related the
Source Second Reprint Second S	
Senarce () Auto Cathologuesia () Sunnamed - it enviruntes agas cuistitation	
ex33466 fores cécel temples agrirer yest	
Opplay Secondar Partire semantics 0, 81-825 antis 4427 894-99311255544	
Deploy Ref: Countremann (fr	
Successful - 8 minutes ago	

Pipeline Successful

You can click on the details link, and it takes you to the Cloud Formation page. You can see the events.

Bits how and and a start of the formation of the formatio						
Addrew 	Q, Filter by stalk name		Stack info	Outputs Parameters 1	Template Change sets	
Control Stands Texadamp Lagical D Status Data result In-data provide In-status * Lagical D Status Data result In-data provide In-status * Lagical D Status Data result In-data provide In-status * Lagical D Status Data result In-data provide In-status * In-status In-status - In-data provide In-status In-status In-status - <td>Active + O View rested</td> <td></td> <td>Events (5)</td> <td></td> <td></td> <td></td>	Active + O View rested		Events (5)			
Nach Threadung Laguar ID Status Status end-supplications 2012-13-171-13431/07-0800 end-supplications OCRATE_CONVENTS - 2012 Lin 171-1341-131/07-0800 Exclusion COLUME_CONVENTS - - 2012 Lin 171-1341-131/07-0800 Exclusion COLUME_CONVENTS - - 2012 Lin 171-1341-131/07-0800 Exclusion COLUME_CONVENTS - - 2012 Lin 171-1341-131/07-0800 Exclusion COLUME_CONVENTS Annual of the termination industrie 2012 Lin 171-1341-131/07-0800 Exclusion Exclusion COLUME_CONVENTS - 1 2012 Lin 171-13441-07-0800 Exclusion COLUME_CONVENTS Status -	< 1.5		Q. Search events			
undurgingenit Social (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	Stacks		Timestamp v	Legical ID	Statue	Status reason
0 0.014/11/044490/008 2022-12-17 134432 UKC 6800 EDMaps © CIXATL_COMPLTE - 2022-12-17 134432 UKC 6800 EDMaps © CIXATL_COMPLTE - 2022-12-17 134432 UKC 6800 EDMaps © CIXATL_COMPLTE - 2022-12-17 134432 UKC 6800 EDMaps © CIXATL_OMOSTIS Analysis 2022-12-17 134432 UKC 6800 EDMaps © CIXATL_OMOSTIS - 2022-12-17 134432 UKC 6800 end dphyment © CIXATL_OMOSTIS Use Instande	ecr-deplayment		2022-12-17 13:44:57 UTC-0600	esr-deployment	O CREATE, COMPLETE	
2012-15-07 134.64 00C 4000 EOMyre © CELTL_7_MIDDLSS Result cancel in Initial 2012-15-07 134.64 00C 4000 EOMyre © CELTL_7_MIDDLSS - II 2012-15-07 134.64 00C 4000 eor-diplayment © CELTL_7_MIDDLSS Use Initialed	CHEATE_COMPLETE		2022-12-17 13:44:55 UTC-0600	ECRRepo	O CREATE_COMPLETE	
202-15-17 14.451 0C-600 K0Mpa @COULT_A_VEDUCISS - II 202-15-17 13.454 0C-600 en-deplayment @COULT_A_VEDUCISS Van Initianal			2022-12-17 15:44:54 UTC-0600	ECRAepo	O CHEATE, IN, PROCRESS	Resource creation initiated
II 202-13-17/13448/07C4000 en-deployment OC4581_7X/m004855 User Initiand			2022-12-17 15:44:52 UTC-0800	ECRAspo	@ CREATE_IN_PROCAESS	*
		11	2022-12-17 13:44:48 UTC-0600	eur-deployment	CREATE IN PROCAESS	User Initiated
		н.	2022-12-17 13-44-48 UTC-0600 2022-12-17 13-44-48 UTC-0600	ECRAço eo-deployment	O CREATE IN PROCRESS	- User initiated

Cloud Formation

You can see the repository created below.

Priv	ate repositori	es (1)			0	View publicammands	Delete	Actions +	Create reposits
Q,	rangenpasitional								C1.5
	Repository		URI	Created at	*	Tag immutability	Scan frequency	Encryption type	Pull through cashe
	-the		Ø 864227929192.dkr.ectus-east- 1.amaconaws.com/webapp	December 17, 2022, 00)	13:44:54 (UTC-	Enabled	Scan on push	AE5-254	mattive

ECR Created

Let's push the Docker Image

Push com	amands for webapp
macOS /	Linux Windows
Make sure ti Started with	hat you have the latest version of the AWS CLI and Docker installed. For more information, see Getting hamazon ECR [2].
Use the follo methods, inc	wing steps to authenticate and push an image to your repository. For additional registry authentication cluding the Amazon ECR credential helper, see Registry Authentication 🔀.
1. Retrieve Use the A	an authentication token and authenticate your Docker client to your registry. W5 CLI:
6 aws	ecr get-login-passwordregion us-east-1 docker loginusername AWSpassword-stdin 227929192.dkr.ecr.us-east-1.amazonaws.com
(Conied	eceive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
scratch :	Docker image using the following command. For information on building a Docker file from see the instructions here 🖄. You can skip this step if your image is already built:
🗇 doc	ker build -t webapp .
3. After th	e build completes, tag your image so you can push the image to this repository:
🗇 doc	ker tag webapp:latest 864227929192.dkr.ecr.us-east-1.amazonaws.com/webapp:latest
4. Run the	following command to push this image to your newly created AWS repository:
do do d	ker push 864227929192.dkr.ecr.us-east-1.amazonaws.com/webapp:latest
	Close
	L

Let's run these commands to push the docker image to the ECR we just created. We will automate this step as well in future articles.

aws ecr get-login-password --region us-east-1 | docker login --username AWS -password-stdin 864227929192.dkr.ecr.us-east-1.amazonaws.com docker tag webapp:latest 864227929192.dkr.ecr.us-east-1.amazonaws.com/webapp:v1 docker push 864227929192.dkr.ecr.us-east-1.amazonaws.com/webapp:v1

You can see the image pushed to the ECR.

hazon	ECR > Reposi	tories	> webapp								
eb	арр									V	iew push
Ima	ges (1)								(C Dela	te][
Q	Search artifacts										
	Image tag	v	Artifact type	Pushed at	•	Size (MB)	٧	Image URI	Digest	Scan status	Val
	v1		Image	December 17, 2022, 14:03:43 (UTC-08)		67.39		CP URI	🖞 sha256:29b7f3fe24bf1dfd3683079f52bce98	Complete	≜ (det

ECR Image

Setup CI/CD Pipeline with AWS Code Pipeline for AWS Apprunner Access the CodePipeline dashboard below and click on the button Create pipeline.



Create Pipeline

You can name the pipeline anything and you can select the service role created above or you can choose to create a new service role and the above policies to the role.

Step 1 Choose pipeline	Choose pipeline settings	de .
Step 2	Pipeline settings	
Add source stage	Pipeline name Enter the pipeline name. You cannot edit the pipeline na	me after it is created.
Add build stage	app_runner_deployment_pipeline	
Step 4	No more than 100 characters	
Add deploy stage	Service role	
Rep 5 Review	New service role Create a service role in your account.	Existing service role Cheose an existing service role from your account
	Role ARN	
	Q. arn:aws:iam::864227929192:role/service-ro	ole/AWSCodePipelineServiceRole-u X
	Advanced settings	
		Cancel Next

Choose pipeline settings	Add build stage into
Step 2 Add source stage	Build - optional
Step 3 Add build stage	Build provider This is the test of your build project. Provide build antifact details like operating system, build spec file, and output file names.
Step 4 Add deploy stage	¥.
Step 5 Review	Cancel Previous Skip build stage Next

Skip Build Stage

You can't skip the deployment stage. I have chosen AWS CloudFormation as a Deployment provider. Make sure you select the right template from the repository.

Step 1 Choose pipeline settings	Add deploy stage 🖦	
Step 2 Add source stage Step 3	You cannot skip this stage Pipelines must have at least two stages. Your secon provider for either the build stage or deployment st	d stage must be either a build or deployment stage. Choose a age.
Add build stage Step 4 Add deploy stage	Deploy	
Step 5	Deploy provider	
Review	AWS CloudFormation	provide the configuration details for that provider.
	- Lossenin and a second second	
	Region	-
	Action mode When you update an existing stack, the update is permanent. When the original stack before you choose to execute the change.	you use a change set, the result provides a diff of the updated stack and
	Create or update a stack	
	Stack name If you are updating an existing stack, choose the stack name.	
	Q app-runner-webapp	×
	Template Specify the template you uploaded to your source location. Artifact name File name	Template file path
	fermitation	Courses & all fronts it committee

Deploy Stage

Once you confirm everything and create a pipeline, you can see the pipeline created successfully and in progress status.

app_runner_deployment_pipeline	🕼 Natify 🔻 🛛 Edit 🔰 Stop xxecution 🗌 Clane pipetine 🔤 🔤
O Source In program Pipeline manufation ID 996/07234-6640-4448-91961-6-0414890-078ef	
Searce © An's Conference © In program - And Four	
Disable transition	
Deploy Derrau	
Deploy (Continential II)	
An execution per	

Pipeline in Progress

If everything is successful, you can see the pipeline successful below.

Code Pipeline

You can click on the next button and choose the source where you read the code. I have selected AWS CodeCommit.

iep 2 dd source stage	Source	
tep 3 dd build stage	Source provider This is where you stored your input artiflacts for your pipeline. Choose the provider and then provid (e the convection details.
tep 4	9	<u>.</u>
dd deploy stage	AWS CodeCommit	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
hep 5	Amazon ECR	Previous Next
eview.	Amazon S3	
	Bitbucket	
	GitHub (Version 1)	
	GitHub (Version 2)	
	GitHub Enterprise Server	

Selecting Source

Select the right repository and the branch you want to deploy.

Add source stage into	
Source	
Source provider This is where you stored your input artifacts for your pipeline. Choose	the provider and then provide the connection details.
AWS CodeCommit	*
Repository name Choose a repository that you have already created where you have pu	shed your source code.
Q, app_cf_templates	×
Branch name Choise a branch of the repository	
Q master	×
Change detection options Choose a detection mode to automatically start your pipeline when a	charge occurs in the source code.
Amazon CloudWatch Events (recommended) Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs	AWS CodePipeline Use AWS CodePipeline to check periodically for changes
Output artifact format Choose the output artifact format.	
 CodePipeline default AVS CodePipeline uses the default zip format for artifacts in the pipeline. Boes not include git metadata about the repository. 	Full clone AVS CodePopulare passes metadata about the repository that allows subsequent actions to do a full git clone. Only supported for AVS Cadebuild actions.
	Add source stage in: Source Source Source S

Source Stage

Since we are running the AWS Code Formation to create the AppRunner, we can skip the build stage. We can build the Docker image here and I will update that in future articles.

bandape tale) Balane) app, some, deplayment, specifie app_runner_deployment_pipeline	App harver > Service > WebApp-dev WebApp-dev	A(Sets ¥)
Searce O Searce O Searce O Searce O Searce O Searce O Statistic Searce O Statistic Searce O	Service everyteee Status Secure Secur	Service AML di enzan apportunen and 1862272919 annar/mologi decitor Inde6246566600 Multicole Tener 1662772919 decess and ansames and engine (2) service
Constant standards Constant standard	Creating and C. C. And S.	C Banking Over at Definition Second Sec

Pipeline Successful

You can click on the details link and it takes you to the CloudFormation page. You can see the events.

prunner-deplayment	apprunner-deployment			
🖸 Stacks (2) 🛛 🔿	Stack info Events Resources	Outputs Parameters Te	Delete Update	Stack actions ¥ G
Q, Filter by stock norm				
Active • Vew nettro	Events (11)			
(1)	Q. Search events			
Stacks	Timestamp v	Logical ID	Status	Status reason
apprunter-deployment	2022-12-17 14:24:11 UTC-0900	apprunner-deployment	@ CREATE_COMPLETE	
OCHEATE_COMPLETE	2022-12-17 14:24:07 UTC-0800	WebApp	@ CREATE_COMPLETE	
eur-deployment	2022-12-17 14:19:29 UTC-0800	WebApp	CREATE_IN_PROGRESS	Resource creation Initia
2022-12-17 13-44-48 UTC-0800	2022-12-17 14:19:27 UTC-0600	WebApp	@ CHEATE_IN_PROGRESS	
Construction of the second	2022-12-17 14:19:25 UTC-0800	AccessRole	@ CREATE_COMPLETE	145
	2022-12-17 14:19:25 UTC-0600	InstanceRole	@ CREATE_COMPLETE	
	2022-12-17 14 19:08 UTC-0800	Accessibale	CREATE_IN_PROGRESS	Resource creation Initiat
	2022-12-17 14:19:08 UTC-0800	AccessRale	CREATE, IN, PROGRESS	(*)
	2022-12-17 14:19:08:UTC-0800	InstanceRole	@ CREATE_IN_PROGRESS	Resource creation Initiat
	2022-12-17 14:19:07 UTC-0800	InstanceRole	CREATE_IN_PROGRESS	* -
	2022-12-17 14:19:03 UTC-0800	apprunner-deployment	CREATE_IN_PROGRESS	User Initiated

Cloud Formation Events

Testing the Webapp

We have created App Runner and ECR by running AWS CodePipeline in the above sections. You can click on the Resources section of CloudFormation [1-5].

						-			
G. Filter	ks (2) C	Stack info	i.	Outputs Parame	ters Template Ch	D ange s	ets	Stack a	tions v
Active	View nested	Resources (3)							
	< 1 >	Q. Search resources							
5	tada	Logical 10		Physical 10 v	Type	19	Status		Module
*	pprunner-deployment 022-12-17 14 19 03 UTC-0800 Dicelette: COMPLETE	AccessRole		apprunner-deployment- AccessRole-727QLILF333J	AWS:SAM:Role		@ CREATE_COMPLETE		8
	or-deployment 022-12-17 13-64-66 uTE-0800	InstanceRole		apprunner-deployment- InstanceRole- 1111VDAG6D2588 🔯	AWS:IAM:Role		O CREATE_COMPLETE		2
e	BORNIL COMPLETE	WebApp		amaws:apprumerus-east- 1.864227929192:service/Web App- des/stitcr1ecbd5d246958a3d7 fcafd5ceb04	WWS:AppRumer:Service		© CREATE, COMPLETE		÷

Resources Created

You can see the AppRunner Created and run successfully.

Service name 9	v	Default domain 🛃	Incoming traffic	Created
NebApp-dev		https://jchmy3vat9.us-east-1.awsa	Public	12/17/2022, 10:19
		v	♥ Default domain 2 https://jchmy3vat9us-east-1.avsa	V Default domain (2) Incoming traffic https://jthmy3vet9us-eait-1.avsa Public

App Runner Running Successfully

You can test the webapp with the following URL. https://jchmy3vut9.us-east-1.awsapprunner.com/

hvutikus-east-1.awsapprunner.com				0.8
	MERN	Stack Exa	mple	
ToDo List				
Task		Assignee		
Create a Task		Assignee		
Status:				
To Be Done				
Submit				
Tasks				
Task Id	Task Name	Assignce	Status	
6017056raan7x60566a-0a60	**	rf.	In Pinaress	

App Runner Running Successfully

Summarv

- If you want to deploy your application on the managed platform by selecting the runtime, AWS App Runner is the right choice.
- You can run the whole API with Docker runtime without any worry about the configuration from your side.
- You can dockerize the WebApp and deploy that in the Docker • runtime. The Docker images can be pulled from ECR, etc.
- Amazon Elastic Container Registry (ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.
- AWS App Runner is an AWS service that provides a fast, simple, and cost-effective way to deploy straight from source code or a container image directly to a scalable and secure web application in the AWS Cloud.
- AWS CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories.

Conclusion

In conclusion, AWS App Runner emerges as an exemplary choice for developers seeking to effortlessly deploy applications on a managed platform, emphasizing simplicity and minimal configuration requirements. By leveraging Docker runtime, it facilitates the smooth operation of APIs and the deployment of web applications directly from Docker images sourced from Amazon Elastic Container Registry (ECR). ECR enhances this ecosystem by offering a robust, managed Docker container registry, streamlining the storage, management, and deployment of container images. Furthermore, AWS App Runner's integration with AWS CodeCommit underscores its commitment to providing a secure, scalable, and efficient deployment pipeline. This synergy between AWS services simplifies the deployment process, from source code or container images to a fully scalable and secure web application, underscoring AWS's role in offering cost-effective,

rapid deployment solutions in the cloud.

References

- 1. Official AWS Documentation https://docs.aws.amazon.com/.
- 2. AWS AppRunner Documentation https://docs.aws.amazon. com/apprunner/.
- 3. Bhargav Bachina (2022) How to develop and build MERN Stack https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-mern-stack-9a7a1099624.
- 4. Bhargav Bachina (2022) How to Build MERN Stack for Production https://medium.com/bb-tutorials-and-thoughts/ how-to-build-mern-stack-for-production-1462e70a35cb.
- 5. Cloud Formation Documentation https://docs.aws.amazon. com/cloudformation/.

Copyright: ©2024 Aauti. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.