WEBVTT

00:00:00.000 --> 00:00:02.880

Қайырлы күн, құрметті студенттер!

00:00:02.880 --> 00:00:06.120

Барлығыңызды келесі Демо сессияда көргеніме қуаныштымын.

00:00:06.120 --> 00:00:16.640

Бүгін алдыңғы Демодағы тапсырманы орындаймыз, яғни AWS бұлтында VPC және оның барлық қажетті құрамдастарын жасаймыз.

00:00:16.640 --> 00:00:27.280

Жалғыз айырмашылық мынада, бұл үшін AWS Management Console емес, AWS CLI, яғни AWS бұлтымен жұмыс істеуге арналған командалық жолын қолданамыз.

00:00:27.280 --> 00:00:32.200

Ендеше, бастайық.

00:00:32.200 --> 00:00:34.400

Қандай тапсырма болғанын еске түсірейік.

00:00:34.400 --> 00:00:42.680

North Virginia аймағында AWS аккаунтын жасау қажет болды, бұл us-east-1 VPC.

00:00:42.680 --> 00:00:48.400

VPC - DEMO\_VPC деп аталады, оның берілген CIDR блогы бар.

00:00:48.400 --> 00:00:54.040

VPC-дегі барлық ресурстар бір availability zone-да болады, us-east-1a.

00:00:54.040 --> 00:00:58.560

VPC-де екі subnet болады, біреуі public, екіншісі private.

00:00:58.560 --> 00:01:06.020

Public subnet тек жай атау екенін және оның шын мәнінде public болуы үшін белгілі бір қадамдар жасалуы керек екенін есте ұстаймыз.

00:01:06.020 --> 00:01:13.800

Ол үшін екі бөлек route table болады, біреуі public subnet үшін, екіншісі private subnet-ке арналған.

00:01:13.800 --> 00:01:20.680

Internet gateway-ді VPC-ге қосамыз, NAT gateway баптаймыз және Elastic IP мекенжайымен жұмыс істейміз.

00:01:20.680 --> 00:01:29.200

Барлық жұмыстарды құмсалғышта орындаймыз.

00:01:29.200 --> 00:01:42.560

AWS Academy-де модульдер тізімінің төменгі жағында Sandbox пунктін таба аласыз және оны бассаңыз, зертханалық жұмыспен ұқсас интерфейс болады.

00:01:42.560 --> 00:01:55.400

Мұнда Terminal терезесін ашып, зертханалық жұмысты бастау керек, осылайша уақытша AWS аккаунты жасалады.

00:01:55.400 --> 00:02:02.760

Оны басқаннан кейін біраз күту керек, содан кейін зертханалық жұмыстың күйі Ready күйіне ауысады.

00:02:02.760 --> 00:02:11.280

Бұл қалқымалы терезені жаба аламыз және AWS батырмасын басқаннан кейін AWS Management Console бағытталасыз.

00:02:11.280 --> 00:02:24.200

Барлық командаларды Terminal-дағы зертханалық жұмыстың парақшасында енгіземіз және кейінірек AWS Management Console-де тексереміз.

00:02:24.200 --> 00:02:29.480

Демоның бірінші бөлігін бастаймыз және енді VPC жасаймыз.

00:02:29.480 --> 00:02:33.600

Сондай-ақ, команда жолымен жұмыс істегенде белгілі бір нюанстарды қарастырамыз.

00:02:33.600 --> 00:02:36.760

Ең бірінші команда - VPC жасау.

00:02:36.760 --> 00:02:39.280

Қосымша компоненттерсіз VPC.

00:02:39.280 --> 00:02:43.320

Команданы көріп тұрсыздар, оны бөліп қарастырайық.

00:02:43.320 --> 00:02:50.120

Бірінші бөлігі - AWS кілт сөзі, ол біз AWS CLI пайдалана бастайтынымызды айтады.

00:02:50.120 --> 00:02:54.720

Әрі қарай екінші кілт сөз – бұл сервис атауы.

00:02:54.720 --> 00:02:58.200

Біздің жағдайда, бұл - EC2.

00:02:58.200 --> 00:03:02.880

Үшінші кілт сөз - API немесе операцияның атауы.

00:03:02.880 --> 00:03:05.440

Біздің жағдайда, бұл - Create VPC.

00:03:05.440 --> 00:03:20.160

Содан кейін жасалған ресурсқа тағайындағымыз келетін мәндері бар қос сызықша арқылы тізімделген параметрлерді көресіз.

00:03:20.160 --> 00:03:25.640

Біздің жағдайда біз VPC жасаймыз.

00:03:25.640 --> 00:03:32.440

Бірінші параметр - бұл ең басында суретте көрген мәні бар CIDR блогы.

00:03:32.440 --> 00:03:34.960

Келесі, екінші параметр - аймақ.

00:03:34.960 --> 00:03:39.400

Мұнда North Virginia аймағын, us-east-1 аймағын көрсетеміз.

00:03:39.400 --> 00:03:45.120

Осы команданы теріп, Enter пернесін басқаннан кейін жауапта JSON аласыз.

00:03:45.120 --> 00:03:48.840

Бұл JSON жасалған ресурсты сипаттайды.

00:03:48.840 --> 00:03:52.160

VPC ID тағайындалғанын көресіз.

00:03:52.160 --> 00:04:01.120

Сондай-ақ, кіріс деректеріне енгізген CIDR блогын және басқа метадеректерді көресіз.

00:04:01.120 --> 00:04:07.960

Енді осы VPC шынымен пайда болғанын тексеріп, AWS Management Console-ді ашайық.

00:04:07.960 --> 00:04:13.540

Мұнда сервисті іздеу жолағын пайдаланып, VPC деп енгізуді бастауымыз керек.

00:04:13.540 --> 00:04:19.440

Дұрыс сервиске кіргеннен кейін, дұрыс аймақта екеніңізді тексеріңіз.

00:04:19.440 --> 00:04:25.560

Сол жақ шарлау мәзірінде Your VPCs пунктін пайдалану керек.

00:04:25.560 --> 00:04:36.440

Бірдей VPC ID және біз тағайындаған CIDR блогы бар VPC жасағанымызды көресіз.

00:04:36.440 --> 00:04:38.040

Енді осы VPC-ні жояйық.

00:04:38.040 --> 00:04:43.600

Мен сізге командалық жолдың мүмкіндіктерін көрсеткім келеді.

00:04:43.600 --> 00:04:51.840

VPC жоятынымызды растауымыз керек және VPC шынымен жойылғаны туралы хабарды көруіміз керек.

00:04:51.840 --> 00:05:07.080

Енді команда жолына ораламыз және VPC жасайтын командаға айнымалы мәндерді тағайындаймыз.

00:05:07.080 --> 00:05:11.120

Бірақ қосымша параметрлер қосылатынын ескеріңіз.

00:05:11.120 --> 00:05:13.240

Бұл мәтінге тең болатын output.

00:05:13.240 --> 00:05:17.280

Яғни, ол команда нәтижесін мәтін түрінде қайтарады.

00:05:17.280 --> 00:05:19.440

Ал екіншісі - query, яғни фильтр.

00:05:19.440 --> 00:05:26.520

Жауапта Vpc.VpcId жолындағы мәнді қайтарғымыз келеді.

00:05:26.520 --> 00:05:32.520

Және бұл мәнді vpc\_id айнымалысына тағайындаңыз.

00:05:32.520 --> 00:05:40.320

Әрі қарай, команда аяқталғаннан кейін, VPC ID айнымалысының мәнін көрсетеміз.

00:05:40.320 --> 00:05:52.840

Сіз басқа бірегей VPC ID алғанымызды көресіз, бірақ командада көрсетілген параметрлермен бірдей.

00:05:52.840 --> 00:05:59.320

Енді AWS Management Console-ге оралып, барлығының дұрыс жұмыс істегенін тексерейік.

00:05:59.320 --> 00:06:04.400

CIDR блогы біз көрсеткен блок екенін және VPC ID сәйкес келетінін көреміз.

00:06:04.400 --> 00:06:16.680

Бұл әрекеттің алдыңғысынан айырмашылығы, команда жолында біз VPC ID-ді айнымалы мәнде сақтадық және оны келесі командаларда пайдалана аламыз.

00:06:16.680 --> 00:06:32.080

Бұл әртүрлі ID енгізу қажеттілігін, адам факторын болдырмайды және команда жолы арқылы AWS бұлтымен жұмыс істеуді жеңілдетеді.

00:06:32.080 --> 00:06:43.760

Енді команда жолына ораламыз және осы жерде тегті Name кілтімен, Demo VPC мәнімен толтырып көрейік.

00:06:43.760 --> 00:06:47.240

Ол үшін келесі команданы қолданамыз.

00:06:47.240 --> 00:06:49.400

EC2 сервисін қолданамыз.

00:06:49.400 --> 00:06:54.640

Әрі қарай Create Tags API және келесі параметрлер бар.

00:06:54.640 --> 00:07:00.440

Resources параметрі үшін VPC ID айнымалы мәнін жіберіп жатқанымызды ескеріңіз.

00:07:00.440 --> 00:07:10.920

Ал Tags параметрінде Name кілті бар тег үшін Demo VPC мәнін тағайындағымыз келетінін белгілі бір форматта жеткіземіз.

00:07:10.920 --> 00:07:18.600

Осы команданы іске қосқаннан кейін, ол қатені қайтармаса, бұл команда сәтті орындалғанын білдіреді.

00:07:18.600 --> 00:07:30.040

Енді, егер AWS Management Console-ге оралсақ, VPC үшін Name бағанының толтырылғанын көреміз, оның мәні Demo VPC.

00:07:30.040 --> 00:07:39.440

Жақсы, команда жолына қайта ораламыз және осы жерде мен назарыңызды келесіге аударғым келеді.

00:07:39.440 --> 00:07:46.680

Ресурста тегтердің болуы сол ресурстар туралы метадеректерді фильтрлеп, сұрауға мүмкіндік береді.

00:07:46.680 --> 00:07:58.720

Мысалы, VPC туралы ақпаратты Name тегі және Demo VPC мәні бойынша фильтрлей аламыз.

00:07:58.720 --> 00:08:07.760

Бұл әрекетті орындау үшін, Describe VPCs API қолданамыз және filters деп аталатын белгілі бір форматтағы параметрді енгіземіз.

00:08:07.760 --> 00:08:11.240

Нәтижесінде JSON қайтарылғанын көресіз.

00:08:11.240 --> 00:08:20.760

Жалғыз VPC кілті бар, ол JSON массиві және әрбір JSON белгілі бір ресурсты сипаттайды.

00:08:20.760 --> 00:08:33.000

Фильтрде белгілі бір VPC көрсеткендіктен, массивте бір JSON қайтарылды және ол біздің VPC сипаттайды.

00:08:33.000 --> 00:08:54.240

Бұл жерде мен команда жолымен жұмыс істегенде бір-бірінен тәуелсіз командаларды жазып қана қоймай, айнымалыларға мәндерді тағайындау және осы айнымалылармен жұмыс істеу туралы ғана емес, бөлек bash скрипттерін де жазуға болатындығына назар аударғым келеді.

00:08:54.240 --> 00:09:03.640

Идея мынада: осы барлық командаларды бір нақты файлға біріктіруге болады және бұл файл белгілі бір тапсырманы орындайды.

00:09:03.640 --> 00:09:06.920

Кішкентай емес, бұл бір команда емес, бірнеше командалардың жиынтығы.

00:09:06.920 --> 00:09:14.480

Осы bash скриптін іске қосқан кезде, осы скрипттегі барлық командалар дәйекті түрде орындалады.

00:09:14.480 --> 00:09:24.840

Осылайша, bash скрипттері арқылы сіз күнделікті жұмысыңыздың белгілі бір бөлігін, Cloud инженердің жұмысын автоматтандыруға болады.

00:09:24.840 --> 00:09:34.160

Онда айнымалыларды да, сонымен қатар күрделірек конструкцияларды, яғни циклдерді, if else конструкцияларды және т.б. қолдануға болады.

00:09:34.160 --> 00:09:40.760

Демоның аясында біз қарапайым опцияны қолданамыз.

00:09:40.760 --> 00:10:05.640

Bash скрипттерін жазу біздің Демо аясына кірмейді, сондықтан біз тәуелсіз командаларды жазамыз, бірақ сонымен бірге белгілі бір мәндерге байланбауға тырысамыз және ресурс ID мәндерін айнымалыларға кейінірек қайта пайдалануға болатындай етіп тағайындаймыз.

00:10:05.640 --> 00:10:12.320

Енді VPC-нің келесі құрамдастарына көшеміз және ең алдымен Internet gateway жасаймыз.

00:10:12.320 --> 00:10:15.520

Ол үшін келесі команданы қолданамыз.

00:10:15.520 --> 00:10:24.040

Мұнда InternetGateway.InternetGatewayId жолындағы мәндерді ig\_id айнымалысына сақтаймыз.

00:10:24.040 --> 00:10:38.000

Команданы орындағаннан кейін ig\_id айнымалысында мәнді көрсеттік және Internet Gateway ID осы жерде көрсетіледі.

00:10:38.000 --> 00:10:45.920

Сондай-ақ, AWS Management Console-де сондай ID тексере аламыз, ол Internet Gateway тізімінде пайда болады.

00:10:45.920 --> 00:10:54.360

Енді осы Internet Gateway үшін тегті Name кілтімен және Demo IGW мәнімен толтыруымыз керек.

00:10:54.360 --> 00:11:07.720

Алдында іске қосқан дәл сол команда іске қосылды, жалғыз айырмашылығы - біз ID Internet Gateway Resources параметріне береміз.

00:11:07.720 --> 00:11:10.680

Оны ig\_id айнымалысында сақтадық.

00:11:10.680 --> 00:11:19.200

Мұны орындағаннан кейін, AWS Management Console-ге оралып, оның сәтті қолданылғанын тексеруге болады.

00:11:19.200 --> 00:11:20.480

Әрі қарай жүрейік.

00:11:20.480 --> 00:11:24.560

Енді жасалған Internet Gateway-ді VPC-ге байланыстыруымыз керек.

00:11:24.560 --> 00:11:30.440

Бұл үшін Attach Internet Gateway деп аталатын API бар.

00:11:30.440 --> 00:11:35.280

Қажетті параметрлер – Internet Gateway және VPC ID.

00:11:35.280 --> 00:11:49.520

Ол екеуі айнымалыларымызда сақталады, сондықтан біз Internet Gateway ID үшін ig\_id айнымалы мәнін және VPC ID үшін vpc\_id айнымалы мәнін көрсетеміз.

00:11:49.520 --> 00:12:04.080

Команданы іске қосқаннан кейін қате болмаса, команда сәтті орындалды деп есептейміз.

00:12:04.080 --> 00:12:06.000

Енді әрі қарай жүреміз.

00:12:06.000 --> 00:12:08.920

Біз subnets жасаймыз.

00:12:08.920 --> 00:12:20.120

Ең алдымен, us-east-1a availability zone шынымен бар-жоғын тексеруіміз керек.

00:12:20.120 --> 00:12:38.380

Шынында, бұл анық, өйткені аймақтың жүйелік атауы us-east-1 екенін білеміз, содан кейін availability zone, мысалы, олар үшеу болса, онда олар алфавиттік тәртіпте белгіленетінін білеміз.

00:12:38.380 --> 00:12:42.360

Бұл us-east-1a, 1b және 1c.

00:12:42.360 --> 00:12:54.640

Бұл команданың идеясы бағдарлама ішіндегі белгілі бір аймақ үшін бар availability zone сұрауға болатын еді.

00:12:54.640 --> 00:13:11.080

Бұл жағдайда value ретінде us-east-1a емес, us-east-1 мәнін көрсетесіз, содан кейін жауап ретінде сізге әрбір availability zone үшін ақпаратты қамтитын JSON бар availability zone массиві қайтарады.

00:13:11.080 --> 00:13:25.160

Бізге қажет availability zone бойынша фильтрлеп, жауап қайтарылғандықтан, осы availability zone жазылуы дұрыс екенін өзіміз үшін тағы да растадық.

00:13:25.160 --> 00:13:27.680

Енді public subnet жасаймыз.

00:13:27.680 --> 00:13:37.460

Мен команданың бөліктері туралы айтпаймын, өйткені идеясы алдыңғы командалардағыдай.

00:13:37.460 --> 00:13:40.520

Ал алдыңғы командаларды біз егжей-тегжейлі талқылаған болатынбыз.

00:13:40.520 --> 00:13:49.440

Мұнда Create Subnet API қолданамыз, қажетті параметрлерді енгіземіз: availability zone, CIDR блогы және VPC ID.

00:13:49.440 --> 00:13:55.320

Содан кейін осының бәрін айнымалыға тағайындаймыз.

00:13:55.320 --> 00:14:00.280

Бізге Subnet.SubnetId жолындағы мән қажет.

00:14:00.280 --> 00:14:04.640

Бірден public\_subnet\_id айнымалы мәніндегі мәнді тексереміз.

00:14:04.640 --> 00:14:07.040

Бір ID бар.

00:14:07.040 --> 00:14:08.040

Жақсы.

00:14:08.040 --> 00:14:09.040

Әрі қарай жүрейік.

00:14:09.040 --> 00:14:15.120

Енді осы subnet үшін Name кілтімен тегті толтыруымыз керек.

00:14:15.120 --> 00:14:18.520

Demo public subnet мәнін толтырыңыз.

00:14:18.520 --> 00:14:20.080

Келесі қадамға көшейік.

00:14:20.080 --> 00:14:25.520

Енді public subnet-те NAT Gateway жасау керек.

00:14:25.520 --> 00:14:30.600

Ол үшін ең алдымен Elastic IP address жасау керек.

00:14:30.600 --> 00:14:35.240

Elastic IP address жасау үшін бөлек API бар.

00:14:35.240 --> 00:14:37.800

Ол Allocate address деп аталады.

00:14:37.800 --> 00:14:52.280

Мұнда қосымша параметрлерді көрсетпейміз, тек JSON ішіндегі мәнді AllocationId жолының бойымен eip айнымалысына сақтаймыз.

00:14:52.280 --> 00:14:58.200

Сіз eip айнымалысында Elastic IP address ID пайда болғанын көресіз.

00:14:58.200 --> 00:15:02.400

Енді әрі қарай жүре аламыз, NAT gateway жасаймыз.

00:15:02.400 --> 00:15:06.920

Бұл жағдайда Create NAT gateway API қолданамыз.

00:15:06.920 --> 00:15:25.560

Мұнда Subnet ID, Allocation ID параметрлерін береміз және мәнді ngw\_id параметрінде Natgateway.NatgatewayId жолында сақтаймыз.

00:15:25.560 --> 00:15:38.600

Келесі қадам, subnet-ті public ету үшін, route table жасап, осы route table-де қосымша параметрлерді жасауымыз керек.

00:15:38.600 --> 00:15:56.680

Сонымен, route table жасау үшін Create route table API қолданамыз, VPC ID параметрлерін енгіземіз және нәтижесінде JSON мәнін public\_rt айнымалысында RouteTable.RouteTableId жолы бойымен сақтаймыз.

00:15:56.680 --> 00:16:00.040

Айнымалыға белгілі бір мән жазылғанын көреміз.

00:16:00.040 --> 00:16:01.040

Жақсы.

00:16:01.040 --> 00:16:18.160

Енді келесі қадам тегті Name кілтімен толтыру болып табылады, мән Public Subnet Route Table болады және енді Route table үшін routing rule қосамыз.

00:16:18.160 --> 00:16:31.680

Естеріңізге сала кетейік, Destination ретінде Интернет пайдаланатын және Target ретінде Internet gateway пайдаланатын routing rule қосуымыз керек.

00:16:31.680 --> 00:16:37.240

Тиісінше, Destination, CIDR блогы мен gateway ID параметрлерін толтырамыз.

00:16:37.240 --> 00:16:51.040

Осының барлығын орындағаннан кейін Return кілті true мәнімен толтырылған JSON түрінде жауап аламыз.

00:16:51.040 --> 00:17:03.880

Қазір барлық қажетті routing rule жасадық, route table жеке тәуелсіз ресурс екенін ұмытпаймыз және оны пайдалану үшін оны subnet-ке байланыстыру керек.

00:17:03.880 --> 00:17:14.560

Әрбір subnet-тің онымен байланыстырылған тек бір route table болуы мүмкін екенін есте ұстаймыз, бірақ кез келген route table-ді бірнеше subnet пайдалана алады.

00:17:14.560 --> 00:17:24.040

Жарайды, келесі Associate route table API қолданамыз, мұнда route table және subnet ID параметрлерін көрсетеміз.

00:17:24.040 --> 00:17:37.120

Мұны істегеннен кейін жауабында JSON-да қауымдастық құрғанымызды көреміз, оның ID ресурсы да бар, оны экранда көре аласыз.

00:17:37.120 --> 00:17:44.600

Қате болмаса, операцияның сәтті аяқталғанын білдіреді.

00:17:44.600 --> 00:17:52.880

Біз Демоның үшінші бөлігіне жеттік және private subnet-ке қатысты құрамдастарды жасаймыз.

00:17:52.880 --> 00:17:56.240

Ең алдымен, Subnet ID жасауымыз керек.

00:17:56.240 --> 00:18:06.920

Сіз команданы көресіз, Create Subnet API қолданамыз, availability zone, CIDR блогын, VPC ID көрсетеміз және барлығын private\_subnet\_id айнымалысында сақтаймыз.

00:18:06.920 --> 00:18:12.800

JSON жүйесінен Subnet.SubnetId жолындағы мәнді қолданамыз.

00:18:12.800 --> 00:18:17.480

Содан кейін мән private\_subnet\_id айнымалысында пайда болғанын көреміз.

00:18:17.480 --> 00:18:27.680

Енді жасалған Subnet ID үшін Name кілтінің мәнін толтыру керек.

00:18:27.680 --> 00:18:36.120

Бұл әрекетті орындау үшін Create tags API қолданамыз және resources пен tags параметрлерін енгіземіз.

00:18:36.120 --> 00:18:50.560

Әрі қарай, route table жасауымыз керек, ол private route table болады, Create route table API қолданамыз, VPC ID параметрлерін толтырамыз, бізде vpc\_id айнымалысында мән бар.

00:18:50.560 --> 00:19:09.960

Әрі қарай, оны оңай ажырату үшін осы route table-ге тег қосуымыз керек. Name кілті бар тегтің мәні private subnet route table болады және бұл жерде route table-де екі өзгеріс енгізу керек.

00:19:09.960 --> 00:19:19.640

Біріншісі - Интернетке трафикті біз бұрын жасаған NAT gateway арқылы өткізуге мүмкіндік беретін routing rule енгізу.

00:19:19.640 --> 00:19:23.580

NAT gateway ID ngw\_id айнымалысында сақталады.

00:19:23.580 --> 00:19:34.480

Осы команданы іске қосқаннан кейін Return кілтінде true мәні орнатылған JSON нәтижесін көреміз.

00:19:34.480 --> 00:19:38.400

Бұл routing rule сәтті жасалғанын білдіреді.

00:19:38.400 --> 00:19:43.520

Енді осы route table-ді private subnet-ке байланыстыруымыз керек.

00:19:43.520 --> 00:20:01.000

Ол үшін Associate route table API қолданамыз, private\_rt айнымалысында сақталған route table ID және private\_subnet\_id айнымалысында subnet ID толтырамыз.

00:20:01.000 --> 00:20:12.320

Нәтижесінде JSON түрінде жауапты көреміз, онда Association ID кілтін кейбір ID енгіземіз, ол операцияның сәтті аяқталғанын көрсетеді.

00:20:12.320 --> 00:20:13.480

Әрі қарай жалғастырайық.

00:20:13.480 --> 00:20:21.680

Енді AWS Management Console-де барлық қажетті ресурстардың жасалғанын тексеруіміз керек.

00:20:21.680 --> 00:20:25.960

Біріншісі - CIDR блогымен сәйкес келетін Demo VPC.

00:20:25.960 --> 00:20:30.080

Әрі қарай, екі subnet бар, public subnet және private subnet.

00:20:30.080 --> 00:20:41.400

Public subnet-те Интернетке трафикті Internet gateway арқылы бағыттайтын қосымша routing rule бар және private subnet бар, онда Интернетке трафик NAT gateway арқылы өтеді.

00:20:41.400 --> 00:20:43.720

Мұнда бәрі дұрыс.

00:20:43.720 --> 00:20:44.720

Әрі қарай жүрейік.

00:20:44.720 --> 00:20:57.160

Екі route table бар, біреуі public subnet үшін және ол онымен байланысты, екіншісі private subnet үшін және оған байланысқан.

00:20:57.160 --> 00:21:03.400

Әрі қарай, Demo VPC-ге қосылған бір Internet gateway бар.

00:21:03.400 --> 00:21:22.680

NAT gateway пайдаланатын Elastic IP address бар және сіз көріп отырған NAT gateway өзі, оның Elastic IP address болып табылатын primary public IP мекенжайы бар.

00:21:22.680 --> 00:21:34.560

Осымен осы тапсырма аясында қажет әрекеттердің барлығын аяқтадық және командалық жолын пайдаланып, дәл сол бірдей ресурстарды жасадық.

00:21:34.560 --> 00:21:36.800

Осымен Демо сессия аяқталды.

00:21:36.800 --> 00:21:43.240

Сіз AWS CLI сервисін жақсырақ түсіндіңіз деп үміттенемін.

00:21:43.240 --> 00:21:45.540

Сұрақтарыңыз болса, жазыңыз.

00:21:45.540 --> 00:21:47.560

Назарларыңызға рахмет.

00:21:47.560 --> 00:21:51.840

Келесі белсенділіктерде кездескенше.