Қайырлы күн, құрметті студенттер! Осы зертханалық жұмысты талдаудың келесі сессиясында сіздермен кездесуге қуаныштымын. Бүгінгі зертханалық жұмыстың тақырыбы – AWS Lambda. Ендеше, бастайық.

Жұмысты бастау үшін біз AWS Management Console-ге жетуіміз керек. Ол үшін зертханалық жұмыстың негізгі парақшасында AWS батырмасын басыңыз.

Осыдан кейін біз AWS Management Console-дің негізгі парақшасына өтеміз. Сонымен қатар, жоғарғы оң жақта "уақытша пайдаланушы" деп тұрғанын көруге болады. Енді біз сервистерді іздеу жолағына Lambda деп енгіземіз және шығарылымда сіз бізге керек сервисті көресіз. AWS Lambda негізгі парақшасына өткеннен кейін қате туралы хабарлама шығады, бірақ біз оған назар аудармаймыз, өйткені нақты AWS аккаунтында мұндай хабарлама болмайды. Әрі қарай, сол жақ навигациялық мәзірден "Functions" қойындысын таңдап, негізгі парақшасында "Create Function" батырмасын басу керек. Зертханалық жұмыстың аясында біз жаңа функция жасаймыз, сондықтан Author from Scratch опциясын таңдаймыз. Төменде функцияның атын енгізіңіз, ол myStopinator деп аталады. Runtime ретінде, яғни орындау ортасы, Python 3.8 нұсқасын таңдаңыз. Төменде осы функция үшін рөл параметрлері бар, біз бар рөлді қолданамыз, яғни тізімнен myStopinatorRole таңдаңыз. Бұл рөл біз үшін арнайы жасалған, ол Lambda үшін қол жетімді, оның EC2 инстанстарын тоқтатуға құқығы бар. Барлық қажетті параметрлерді енгізгеннен кейін біз төменгі оң жағындағы Create Function батырмасының басамыз. Бізді осы функциямыз бар парақшаға бағыттайды және функция сәтті жасалғандығы туралы тиісті хабарлама пайда болады.

Біздің келесі қадамымыз – триггер-ивент жасау. "Add Trigger" батырмасын басу арқылы біз дереккөз таңдайтын парақшаға бағытталамыз. Дереккөз, яғни триггер Event Rule болады, сондықтан біз event деп енгізе бастаймыз және ашылмалы мәзірде EventBridge (CloudWatch Events) опциясын таңдаймыз. Бізде бар ивентті таңдау немесе жаңасын жасау мүмкіндігі пайда болады. Жаңа ивент жасау (Create a new rule) опциясын таңдап, оны "everyMinute" деп атайық. Түрі ретінде біз Schedule Expression, яғни кесте бойынша қайталанатын оқиғаларды таңдаймыз. Біздің жағдайда бұл rate (1 minute) болады. Бұл ивент біздің лямбданы минут сайын қоздыруы үшін қажет. Барлық параметрлерді енгізгеннен кейін төменгі оң жақтағы Add батырмасын басу керек. Оны басқаннан кейін сіз біздің функциямыз бар парақшаға қайта бағытталасыз, онда біздің триггер сәтті жасалғандығы және біздің парақшада лямбда-функциясын іске қосатын бір триггер, EventBridge триггері бар екендігі туралы хабарлама болады.

Енді біздің функциямыздың кодын өзгерту керек. Ол үшін төменге айналдырып, "Code" қойындысында арнайы код редакторын ашамыз. Мұнда біз lambda\_function.py таңдаймыз және осы файлдың кодын ашамыз. Сол кодтың ішіндегі барлық нәрсені тазалап, зертханалық жұмыста берілген код үлгісін енгізу керек. Мұнда кодқа кейбір өзгерістер енгізу қажет болады, атап айтқанда аймақты ауыстыру. Біздің жағдайда, бұл North Virginia, сондықтан біз "us-east-1" жүйелік атауын енгіземіз. Әрі қарай, EC2 инстанстың идентификаторын (ID) енгізу керек. Осы сервисқе өтеміз, одан кейін сол жақ навигация мәзірінде Instances қойындысына өтеміз. Бұл ретте сіз дұрыс аймақта, атап айтқанда North Virginia аймағында екеніңізге көз жеткізуіңіз керек. Осыдан кейін сіз инстанстардың тізімінде бір немесе екі инстансты көресіз, бізге instance1 деп аталатын инстанс қажет. Оның статусына назар аударыңыз, ол running күйінде болуы керек. Әрі қарай, біз осы инстансты таңдаймыз, қосымша ақпаратта Details қойындысына өтіп, Instance ID-ді көшіреміз. Оны көшіргеннен кейін лямбда-функциясына ораламыз және үшінші жолда алдыңғы парақшадан көшірген Instance ID мәнін енгіземіз. Осыдан кейін кодты сақтау керек. Мұны Ctrl+S батырмалар тіркесімін басу арқылы немесе File мәзір элементінен Save опциясын таңдау арқылы жасауға болады. Лямбда-функцияның жаңа коды жұмыс істей бастауы үшін оны Deploy батырмасын басу арқылы орнату керек және бірнеше секундтан кейін функцияның коды сәтті жаңартылғандығы туралы хабарлама пайда болады.

Сақталған код бойынша қысқаша шолу жасап өтейік. Бірінші жолда біз SDK импорттаймыз. Python үшін SDK – бұл Boto3, сондықтан мұнда import boto3 болады. Әрі қарай, екінші және үшінші жолдар айнымалыларды жариялау болып табылады, атап айтқанда аймақ және біздің инстанстарымыз туралы ақпарат. Бірінші жағдайда – бұл мәтіндік деректер түрі, ал екінші жағдайда – string-тардан тұратын массив. Осылайша, егер бір емес, бірнеше инстанстарды тоқтату қажет болса, осы массивте қосымша мәндерді беруге болады. Әрі қарай, төртінші жолда біз EC2 сервис үшін Boto3 клиентін бастаймыз, яғни, біз Boto3.client-ті іске косамыз, қандай сервис қажет екенін көрсетеміз, біздің жағдайда бұл EC2, және қосымша біздің аймақтың атауын көрсетеміз. Мұның бәрін біз EC2 айнымалысына тағайындаймыз. Келесі – AWS ішіндегі Python кодының стандартты құрылысы, атап айтқанда lambda\_handler функциясы. Кіру үшін ол event пен context қабылдайды. Бұл жүйелі түрде толтырылатын екі айнымалы, яғни біздің лямбда-функциясы шақырылған кезде, кейбір деректерді тасымалдау кезінде бұл деректер event айнымалысында болады. Сондай-ақ, осы кодтың орындалуы туралы кейбір мета-ақпарат бар, бұл ақпараттың барлығы context айнымалысында сақталады. Бұл функцияның ішінде жетінші жолда біз ec2.stop\_instances клиентін қолданамыз, біз бұл action-ді шақырамыз. Сондай ақ, AWS-те белгілі бір әрекеттерді орындау үшін үш нұсқаны қолдануға болатынын есімізде бар. Біріншісі – консольдің өзін пайдалану, яғни біз консоль арқылы өз бетінше тоқтата аламыз. Екіншіден, біз AWS CLI-ді қолдана аламыз, яғни, командалық жол, онда біз Stop Instances командасын шақырамыз және осы командамен біз инстанстарды тоқтатамыз. Үшінші нұсқа – SDK пайдалану, бұл бұлтта белгілі бір әрекеттерді жасайтын жоғары деңгейлі кодты жазу. Біздің жағдайда біз инстанстарды тоқтатамыз. Ішінде жақша ішінде қосымша параметрлер жіберіледі, біздің жағдайда біздің instances айнымалысына тең instanceID параметрі, яғни біз instanceID тізімін кіріске жібереміз. Әрі қарай, сегізінші жолда біз инстанстар туралы ақпаратты хабарлама ретінде шығарамыз.

Бұл Лямбда кодының ең қарапайым мысалы болды және сіз болашақта жазатын кез келген код осы мысалға ұқсас болады. Сіз SDK-ді жиі қолданатын боласыз, бұл Boto3, Python бағдарламалау тілін қолданған жағдайда, негізі бұл онша қиын емес. Сонымен қатар, біз бұлтта жұмыс істейтін кодты дайындадық және сізге жеке серверді орнатудың немесе қосымша ештеңе орнатудың қажеті болмады. Біз тек кодты жүктеп алдық, оны орнаттық, сақтадық, енді бұл код минут сайын іске қосылады.

Енді "Monitor" қойындысына өтіңіз, онда сіз осы лямбданың іске қосылуына қатысты қандай ақпарат бар екенін көре аласыз. Біз тек үш графикке тоқталамыз. Біріншісі – Invocations, уақыт бойынша біздің лямбда-функцияның іске қосылу саны. Мұнда кішкене сызық көрініп тұр, бұл біздің лямбданың екі рет іске қосылғанын көрсетеді. Әрі қарай, екінші Duration графигі біздің функциямыздың қанша миллисекунд іске қосылғанын көрсетеді, біздің жағдайда бұл 18 миллисекунд. Үшінші график – Error count and success rate, ол лямбда-кодының сәтті және сәтсіз орындалуының пайыздық қатынасын көрсетеді. Біздің жағдайда қызыл индикатор 0-де, яғни біздің барлық іске қосуларымыз сәтті өтті, ал жасыл 1 деңгейінде, яғни 100%. Керемет! Енді біздің инстанстарымыз бар парақшаға оралайық. Біз кодтың бірнеше рет іске қосылғанына көз жеткіздік, енді инстанстар тізімін жаңартып көрейік. Жаңарту батырмасын басу арқылы сіз instance1-дің инстанс күйі Running-тен Stopped-ке өзгергенін көресіз, яғни біздің код жұмыс істеп, инстансты сәтті тоқтатты.

Осымен біздің зертханалық жұмысымыз аяқталады. Барлық жүйелерден дұрыс шығу керек екенін еске саламын. Ең алдымен, AWS Management Console-дан шығуымыз керек. Ол үшін жоғарғы оң жақта пайдаланушы атын және ашылмалы мәзірде "Sign Out" батырмасын басыңыз. Әрі қарай, зертханалық жұмыс парақшасында біздің AWS аккаунт жойылуы үшін зертханалық жұмысты тоқтату керек. Ол үшін End Lab батырмасын басып, зертханалық жұмысты аяқтағанымызды растаймыз және "You may close this message box now" хабарламасын күтеміз. Осыдан кейін сіз барлық терезелерді жауып, жүйелерден сәтті шыға аласыз.

Осымен біз зертханалық жұмысты талдау сессиясын аяқтаймыз. Сіз жаңа нәрсе білдіңіз және AWS Lambda сервисі туралы толық түсінік алдыңыз деп үміттенемін. Рахмет, келесі белсенділіктерімізде кездескенше!