Қайырлы күн, құрметті студенттер! Сіздерді бүгінгі дәрісте көргеніме қуаныштымын. Бүгін біз Networking пен Content Delivery туралы сөйлесеміз. Сонымен, бастайық.

Бүгінгі дәріс алты бөлімге бөлінген. Бірінші бөлімде біз желілердің негіздері туралы сөйлесеміз және негізгі ұғымдарды, яғни Networking basics-ті қарастырамыз. Әрі қарай, келесі үш бөлімде біз Amazon Virtual Private Cloud сервисі туралы сөйлесеміз, көп жағдайда бұл сервис сізге Amazon VPC деп кездеседі. VPC networking және VPC security тақырыптарын қозғаймыз. Соңғы екі тақырып – екі бөлек сервис, біріншісі – Amazon Route 53 сервисі, екіншісі – Amazon CloudFront сервисі.

Біз бірінші бөлімді бастаймыз, бұл желілердің негіздері. Біз сізге келесі бөлімдерді, атап айтқанда AWS жүйесінде желі құру туралы айтатын бөлімді, әсіресе Amazon VPC сервисін түсінуді жеңілдету үшін желі негіздерін қарастырамыз. Желі дегеніміз не? Желі – бұл ақпарат алмасу үшін бір-бірімен байланысқан екі немесе одан да көп клиенттік машиналар. Желіні subnets деп аталатын логикалық бөліктерге бөлуге болады. Машиналарды бір-біріне қосу үшін бізге желілік жабдық қажет. Switch немесе router желілік жабдық ретінде әрекет ете алады. Желідегі әрбір машинаның белгілі бір желідегі құрылғыны анықтайтын бірегей IP-адрес болады.

IP-адрес – төрт бөліктен тұратын ондық санақ жүйесіндегі сандық белгі. Содан кейін, сіз көріп отырғандай, машина екілік санақ жүйесіне түрлендіріледі, осылайша, бағдарлама осы IP-адреспен ең төменгі деңгейде жұмыс істейді. Біз оны ондық санақ жүйесінде көреміз, ал мысал ретінде сіз 192.0.2.0 көріп отырсыз. IP-адрес ішіндегі әрбір бөлім нөлден 255-ке дейінгі мәндерді қабылдай алатын қандай да бір сандар, яғни 256 бірегей мән. 256 – 2-нің 8-ші дәрежесі, ендеше екілік санақ жүйесінде нөлден 255-ке дейінгі сандарды белгілеу үшін 8 бит бөлу керек. 8 рет нөлден (00000000) бастап, 8 рет бірмен (11111111) аяқталатын сандар. Осылайша, бинарлық форматтағы IP-адрес 8-ді 4-ке көбейткенде, яғни 32 битке тең.

Біз IP-адрес 32 биттен тұрады деп айттық. Бұл IPv4-адрес деп аталады. Сондай-ақ, бізде IP-адрестің тағы бір түрі – IPv6-адрес бар, ол 128 биттен тұрады, және ол белгілі бір желі үшін одан да ерекше IP-адрестер жасай алады. IPv6-адресі 8 топ түрінде ұсынылады. Әр топта әріптер мен сандар болуы мүмкін. Біз бұл адресті он алтылық санақ жүйесінде көреміз және әр топ 16 биттен тұрады және нөлден бастап FFFF-ке дейінгі мәнді қабылдай алады. Тағы да айтып өтейін, бұл он алтылық санақ жүйесіне тән. Жұмыс барысында сіз тікелей IPv6-адрестерді жиі кездестірмейсіз. Сіз негізінен IPv4-адреспен жұмыс жасайсыз. Сонымен, оны толығырақ қарастырайық.

Жергілікті желіде мүмкін IPv4-адрес мәндерінің тізімін анықтау үшін біз CIDR деп аталатын блокты қолданамыз. CIDR – толығымен Classless Inter-Domain Routing деп аталады. Ол келесі форматта жазылады: бұл бастапқы, яғни IPv4 форматтағы ең алғашқы IP-адрес, содан кейін слэш және 0-ден 32-ге дейінгі мәнді қабылдай алатын қандай да бір сан. Бұл бізде қанша бит бекітілгенін көрсетеді. IPv4-адресі 32 биттен тұрады. Сондықтан, егер CIDR блокта 24 жазылғанын байқасақ, бұл 32 биттің 24-і бекітілген және оны өзгерту мүмкін емес дегенді білдіреді. Осылайша, қалған 8 бит өзгеруі мүмкін. Бұл 8 бит – бірегей IP-адрестердің саны 2-нің 8 дәрежесі 256-ға тең болмақ. Осылайша, слайдта көрсетілген біздің CIDR блокта IP-адрес нақты 256 бірегей мәнді қабылдай алады. Бұл желіге тиесілі алғашқы IP-адрес – 192.0.2.0. Осылайша, өзгере алатын соңғы бөлік, 0-ден әрі қарай 0.1.2.3.4 деп жүре береді, яғни ең соңғы IP-адрес 192.0.2.255 болады. Жақсы. Бұл сіздерге түсінікті деп ойлаймын және материалды бекіту үшін біз келесі тапсырманы орындайық.

Бізде бірінші бөлігі бірдей CIDR блок бар деп елестетіп көрейік, яғни 192.0.2.0, ал 24 орнына слэш 16 болады. Бұл видеоны тоқтатып, осындай CIDR блогына сәйкес осы желіде қанша бірегей адрестер болатынын есептеп көрейік. Жарайды, менің ойымша, барлығыңыз жауабын таптыңыз. Дұрыс жауап – 65536. Біз бұл санға қалай жеттік? Біздің CIDR блогымыз 192.0.2.0/16 екенін айттық. Бұл 32 биттік IP-адрестің 16-сы бекітілгенін білдіреді. Бұл дегеніміз, бізде flexible 16 бит барын, яғни олар өзгере алатынын білдіреді. Біздің 16 бит – бұл 2-нің 16 дәрежесіне тең бірегей IP-адрестер дегенді білдіреді. Егер біз 2-нің 16 дәрежесін санасақ, бұл 65536. Жарайды, менің ойымша, енді сізге бәрі түсінікті болды.

Тағы екі шекаралық жағдайды қарастырайық. Бізде 192.0.2.0/32 типтегі CIDR блок бар деп елестетіп көрейік. Тағы бір тапсырманы орындап, осы CIDR блокта қанша IP-адрес бар екенін есептеуге тырысайық. Сіз видеоны тоқтата тұруыңызға болады. Ал, мен сіздердің әрқайсыларыңыз дұрыс есептеп, дұрыс жауап ала алды деп сенемін. Дұрыс жауап – 1. Мұнда /32 деп аяқталатын CIDR блок бір машинаны анықтайды. Мұны қалай есептейсіз? /32 деп айтқан кезде, бұл бізде 32 бит бекітілген дегенді білдіреді. Біздің IPv4-адресімізде тек 32 бит болғандықтан, бізде қанша бірегей адрестер бар екенін есептеу үшін біз 32-ден бекітілген 32 битті алып тастаймыз. 32-32=0 болып шығады, ал 2-нің 0 дәрежесі 1-ге тең. Осылайша, сіз Amazon-да жұмыс істеген кезде, AWS Management Console-де белгілі бір IP-адресті көрсету үшін сіз CIDR блок /32 екенін білесіз және сіз өзіңіздің IP-адресіңізді сәйкесінше көрсетесіз.

Екінші шекаралық жағдай, оны тапсырма ретінде бермей, бірден жауап беріп көрейік. Бізде 0.0.0.0/0 CIDR блогы бар десек, бұл жағдайда біз ең алғашқы IP-адрестен бастайтынымызды көреміз, өйткені бізде бекітілген биттердің саны 0 және барлық биттер өзгеруі мүмкін, яғни flexible күйінде тұрады. Бұл бізге осы желіде 2-нің 32 дәрежесіне тең бірегей адрестер бар екенін көрсетеді және бұл бүкіл интернетті білдіреді. Сондай-ақ, AWS Management Console-де сіз бүкіл интернетке кіруге рұқсатты немесе интернетке дейін route бағыттауды немесе интернетке кіруге тыйым салуды көрсету қажет болғанда, CIDR блогы ретінде сіз әрқашан нүкте арқылы бөлінген барлық нөлдер слэш 0 деп көрсете аласыз, сонда бұл Интернет болады.

Келесі қарастырылатын модель, мүмкін сіз оны өткен де шығарсыз, бұл OSI модель, яғни Open Systems Interconnection Model. Бұл модельдің негізгі идеясы – ол деректердің желі арқылы қалай берілетінін концептуалды түрде сипаттайды, яғни application-нан бастап, бірліктер мен нөлдер тізбегі деңгейінде жұмыс істейтін физикалық деңгейге дейін 7 деңгей бар. Application-ның ең жоғарғы деңгейі – HTTP, HTTPS протоколдарын және т.б. пайдалана отырып жұмыс істеген кез. Осылайша, біз белгілі бір сайтқа HTTP request сұраныс жасаған кезде, ол жоғарғы деңгейден ең төменгі деңгейге дейін төмендейді, нәтижесінде бізден нөлдер мен бірліктердің тізбегі біздің алушыға өтеді. Біздің бірліктер мен нөлдеріміздің тізбегі, яғни біздің деректеріміз немесе сұранысымыз алушыға жеткенде, ол осы тізбектің шифрын ажыратып, оқу, түсіну және қандай да бір хабарламамен жауап беру үшін application-нің жоғарғы деңгейіне көтеріледі. Жауап сонымен қатар жоғарғы деңгейден төменге қарай түседі және сізге келген бойда, осы тізбекті оқып шығу үшін сіз ең төменгі деңгейден ең жоғары деңгейге өтіп, сізге жіберілген жауапты бастапқы түрінде оқи аласыз. Бұл ақпарат желінің қалай жұмыс істейтінін, деректердің желі арқылы қалай ағып жатқанын және жалпы алғанда, Amazon VPC сервисінің қалай жұмыс істейтінін түсіну үшін жеткілікті.

Біз дәрістің екінші бөліміне көшеміз, бұл бөлімде Amazon VPC сервисі туралы көбірек білетін боламыз. Amazon VPC – Amazon Virtual Private Cloud деп оқылады. Бұл – бұлтта логикалық оқшауланған желіні орнатуға мүмкіндік беретін сервис. Сіз осы желіні толығымен басқарып, осы желіде ресурстар жасай аласыз. Сонымен қатар, сіз осы желідегі кез-келген теңшелетін параметрлерді қолдана аласыз, сонымен қатар қауіпсіздіктің бірнеше деңгейі бар, бұл Security Group және Network Access Control Lists немесе жиі сіз Network ACLs деп кездестіретін боласыз.

Мұнда сіз VPC бұлтта қалай орналасуы мүмкін екенін түсіндіретін диаграмманы көре аласыз. VPC тек бір AWS аккаунтта орналаса алады. Сонымен қатар, VPC тек бір аймақта толығымен орналасуы керек. Егер сізге бірнеше аймақты пайдалану қажет болса, сізде әр аймақ үшін өз VPC-і болуы керек. Біз аймақ ішінде бірнеше availability zone болуы мүмкін екенін білеміз. Осылайша, бір VPC бірнеше availability zone-да болуы мүмкін.

VPC біз бұрын айтқан CIDR блокпен анықталады. Сондай-ақ, VPC ішінде біз бірнеше логикалық топтарды жасай аламыз, олар subnets деп аталады. Subnets-тер де кішігірім CIDR блокпен анықталады. Және range, яғни subnet-тің IP-адресінің ықтимал мәндері VPC-дің мүмкін мәндеріне кіру керек. Subnets екі түрлі болуы мүмкін. Біріншісі – бұл private, екіншісі – public. Public – интернеттен қол жетімді subnets, ал рrivate бұл интернеттен қолжетімсіз subnets.

Мен VPC-де болуы мүмкін CIDR блоктар мөлшері туралы бірнеше сөз айтқым келеді. Ондағы максималды өлшем – 65536 бірегей IP-адрес, ол /16 CIDR блокқа тең деген сөз. Ең кішкентайлары – 16 IP-адрес. CIDR блокта ол /28 түрінде жазылады, яғни 4 түрі flexible болуы мүмкін, яғни өзгере алады. VPC-ге арналған CIDR блок туралы айтқан кезде, осы CIDR блоктың өлшеміне, сондай-ақ бастапқы IP-адреске мұқият және жауапкершілікпен қарау қажет. Себебі, VPC-де тағайындаған CIDR блокты кейінірек өзгерту мүмкін емес. Сондықтан CIDR блокты өзгерту қажет болса, жаңа VPC жасау керек болады. Егер VPC-де жұмыс істейтін ескі ресурстарыңыз болса, сіз осы ресурстардың барлығын жаңа VPC-дің жаңа CIDR блогына ауыстыруға уақыт, күш, ақша жұмсайсыз және осының бәрін жасайтын адамдар керек болады. Ал бұл өте қымбат болуы мүмкін, сондықтан VPC өлшемін мейлінше артығымен алуды басты ереже деп біліңіз. Сонымен қатар, VPC-лер бір-бірімен қиылыспайтынын әрқашан ұмытпаңыз. Егер VPC әртүрлі AWS аккаунтта болса да, болашақта бір-біріне сәйкес келмейтін CIDR блоктарды жасауды ұсынамын, өйткені AWS-та әртүрлі аккаунтардың VPC-ін бір-бірімен қосуға көмектесетін сервистер бар. Егер бұл VPC-лардың CIDR блоктары қиылысса, сіз оларды біріктіре алмайсыз, сондықтан әрқашан осы тұсқа қатты мән беріңіз.

Келесі мәселе subnet-тер туралы. Жоғарыда айтылғандай, subnet-тің CIDR блогы VPC-дің CIDR блогының бөлігі болуы керек. Subnet-тің максималды өлшемі тұтас VPC-дің өлшеміне тең болуы мүмкін. Осылайша, бұл VPC-де тек бір subnet болады. Егер біз VPC ішіне бірнеше subnet-тер жасасақ, бір VPC ішіндегі CIDR блоктар да бір-біріне сәйкес келмейтініне көз жеткізуіміз керек.

Үлкен не кіші көлемді болсын, CIDR блокты жасаған кезде, AWS бұлтындағы VPC не subnet үшін бе, маңызды емес, AWS жүйелі түрде пайдалану үшін 5 IP-адресті резервке қояды. Сіз CIDR блоктағы range-те ең бірінші IP-адрес network address үшін сақталғанын көре аласыз, екіншісі internal communication үшін, үшіншісі DNS resolution үшін, төртіншісі – болашақта пайдалану үшін сақталған, ал соңғысы network broadcast address ретінде пайдаланылады. Осылайша, /24 өлшемі бар CIDR блок үшін, яғни 256 бірегей адрес, және сіздің пайдалануыңыз үшін қол жетімдісі – 256-5 = 251 IP-адрес.

Қазір тағы бір тапсырма орындайық. Сізде келесідей CIDR блок бар деп елестетіп көріңіз: 10.0.0.0/28. Сұрақ: Сіз пайдалану үшін осы CIDR блок ішінде қанша IP-адрес қол жетімді? Жарайды, менің ойымша, барлығыңыз жауабын таптыңыз. Бірге есептеп көрейік. /28 өлшемі бар CIDR блок 32 биттік IPv4 CIDR блоктағы 28 биттің бекітілгенін көрсетеді, яғни өзгеруі мүмкін биттердің саны 32 минус 28 – 4-ке тең болады. 4 бит үшін қанша IP-адрестерді есептей аламыз. Ол үшін біз шыққан санның дәрежесіне қарай 2-ні санаймыз, біздің жағдайда ол 4, 2-нің 4 дәрежесі 16-ға тең. Жаңа айтқанымыздай, AWS жүйелі түрде пайдалану үшін 5 адресті резервке қояды. Сонымен, 16-дан 5-ті азайтсақ, нәтижесі 11-ге тең. Сонымен жауабы – /28 CIDR блогы үшін біздің пайдалануымызға қолжетімді бірегей IP-адрестің саны 16 емес, 11 болмақ.

Public IP-адрестердің екі түрі бар. Біріншісі – жалпыға ортақ IPv4-адрес және Elastic IP-адрес. Олар қалай ерекшеленеді? AWS бұлтында VPC жасаған кезде, осы VPC ішіндегі барлық инстанстар автоматты түрде private IP address деп аталатын ішкі IP-адреске ие болады. Бірақ мұнымен қатар, VPC жасау кезінде Auto assign public IP address пунктіне құсбелгісін қоюға болады. Осылайша, private IP address-тен басқа, барлық инстанстар динамикалық public IP address болып табылатын public IPv4 address-теріне ие болады. Статикалық жалпыға ортақ IP-адрес қажет болса, бұл жағдайда Elastic IP address пайдалануға болады. Бұл IP-адрес ресурсқа, атап айтқанда инстансқа немесе VPC ішіндегі интерфейске байланыстыра аласыз. Elastic IP address қызметі де ақылы екенін есте ұстаған жөн. Осылайша, бекер төлем жасамау үшін ресурстарға байланбаған барлық Elastic IP address-терін босату ереже болып қалыптасуы керек.

Ал енді Elastic Network Interface туралы бірнеше сөз айтқым келеді. Бұл сіздің VPC-дің ішіндегі инстансыңызға тіркеле алатын жеке ресурс, ол белгілі бір IPv4-адреске қызмет ете алады. Сонымен қатар Elastic Network Interface, әр инстанстың VPC ішінде Default Network Interface болады. Бұл Default Network Interface-ке VPC-дің CIDR блогынан таңдалған Private IPv4-адрес тіркеуге болады. Сіз инстансқа Network Interface-тің белгілі бір санын тіркей аласыз және бұл сан инстанстардың түріне байланысты және ол неғұрлым жоғары болса, соғұрлым үлкен, соғұрлым көп Network Interface-ті белгілі бір инстансқа қоса аласыз.

Ал енді Route table не екенін мұқият қарастырайық. Route table ережелер жиынтығынан тұратын кесте болып табылады, rule немесе routes. Әрбір ереже жөнелту орны мен баратын жері бойынша немесе ағылшын тілінде айтқанда destination and target-пен анықталады. Бізде destination ретінде CIDR блок әрекет етеді, ол біздің subnet-ті сипаттай алады, ал target ретінде белгілі бір ресурс көрсетіледі. Кез-келген Route table-де бірінші, ең негізгі rule бар. Бұл rule біздің барлық subnet-терден local-ды target ретінде көрсетеді. Бұл біздің subnet-тер арасындағы барлық инстанстар бір-бірімен өзара әрекеттесуі үшін қажет.

VPC-де әдепкі бойынша Route table бар, ол Main Route tablе деп аталады. Ол автоматты түрде жасалынып, біздің VPC-мен байланыстырылады. Маңыздысы, барлық subnet-тер бекітілген болуы керек, яғни Route table-дің біреуіне байлануы керек. Бірақ әр Route Table-дің бірден көп subnet-і болуы мүмкін. Сонымен қатар, әр VPC-де бірнеше Route table болуы мүмкін. Осылайша сіз subnet-тер арасындағы барлық routes-теріңізді икемді түрде баптай аласыз.

Біз екінші бөлімді аяқтаймыз. Есте сақтау керек негізгі ойларға шолып өтейік. VPC – AWS бұлтының логикалық түрде оқшауланған бөлігі, онда біз қажетті ресурстарды жасай аламыз. VPC тек бір аккаунтқа және бір нақты аймаққа қолданылады. Бірақ VPC бір аймақтың ішіндегі бірнеше availability zones-тарда орналасуы мүмкін. VPC бір нақты CIDR блокпен анықталады. Біз алдыңғы бөлімде CIDR блоктың не екенін талқыладық. Барлық VPC-ді subnet-тер деп аталатын логикалық топтарға бөлуге болады. Әр subnet VPC-ге кіретін CIDR блоктың бөлігі болуы тиіс CIDR блокпен анықталады. Route table деген ұғым бар, бұл біздің VPC ішіндегі subnet-теріміздің арасындағы трафикті басқаратын ресурс. Әр Route table ішіне кірістірілген Local Route жол бар, оны жою мүмкін емес және әртүрлі subnet-тердегі барлық инстанстардың VPC ішінде бір-бірімен байланысуға мүмкіндігін қамтамасыз етуге көмектеседі. VPC ішіндегі желілік ағындарды икемді баптау үшін Route table-ге қосымша routes, rules немесе ережелер қосуға болады.

Сонымен біз үшінші бөлімге де келіп қалдық. Үшінші бөлімде VPC ішінде ішкі трафикті бағыттауды баптау ерекшеліктері туралы сөйлесеміз. VPC-дің маңызды компоненттерінің бірі Internet gateway болып табылады. Internet gateway интернетті сіздің VPC ішіндегі инстанстарыңызға қосу үшін қолданылады. Сізді public subnet көпшілікке, яғни интернеттен қол жетімді ету үшін Internet gateway қосып, Route Table-ге destination ретінде интернет жүретін бір route қосуымыз керек, яғни 0.0.0.0/0, target ретінде Internet gateway әрекет етеді.

Тағы бір жиі кездесетін жағдай – private subnet ішіндегі инстанстардың интернетке өту қажеттілігі. Сонымен қатар, біз private subnet ішіндегі инстанстарға интернеттен қол жетімділікті болдырмауымыз керек. Ол үшін Network Address Translation gateway қолданылады, сіз оны NAT gateway деп жиі кездестіресіз. Gateway-ді дұрыс орнату үшін алдымен public subnet-ті дұрыс күйге келтіру керек. Біз Public subnet-ті қалай орнату керектігін білеміз. Қайталап өтейін. Сізде public subnet үшін арнайы route table бар деп елестетіп көріңіз және онда сіз бүкіл интернетті destination ретінде және Internet gateway-ді target-і ретінде көрсетесіз. Осылайша subnet-іңіз public болады. Әрі қарай, public subnet ішінде біз NAT gateway жасаймыз. Енді private subnet-ке арналған route table-ге интернеттен route қосу керек, яғни destination – Интернет, ал target – NAT gateway. Осыдан кейін жеке ішкі желідегі барлық инстанстар NAT gateway арқылы интернетке қол жеткізе алады. Сонымен қатар, NAT gateway интернеттен сұрауларды private subnet-ке қайтаруға мүмкіндік бермейді. NAT gateway-ден басқа, NAT instance-ты баптайтын тағы бір мүмкіндігі бар, яғни бөлек EC2 инстансын баптап, онда NAT gateway теңшеледі. Айырмашылығы мынада, егер инстанс істен шықса, оған бірдеңе болса, онда сіздің интернет жоғалады және оны ауыстыру қажет болады. Егер біз NAT gateway туралы айтатын болсақ, бұл Amazon-ның managed service-і, сондықтан барлық ықтимал қателерді, қол жетімсіздікті Amazon қамтамасыз етеді және қандай да бір ресурс істен шыққан жағдайда ол сіз үшін жаңа жұмыс ресурсымен анық түрде ауыстырылады. Осылайша, Amazon NAT instance-ы орнына NAT gateway-ді пайдалану best practice ретінде ұсынады.

Сіз AWS ішінде жүзеге асыра алатын бір қызықты тәсіл – VPC sharing. Бұндағы идея, сіз өзіңіздің VPC ішіндегі subnet-теріңізді басқа VPC аккаунттарымен бөлісе аласыз. Бірақ бұл аккаунт жалпыға ортақ AWS Organizations-ке кіруі керек. Айта кететін бірнеше нюанстар да бар. Бұл сіз shared subnet ішіндегі осы ресурстардың барлығын көре аласыз және өзгерте аласыз, бірақ сіз осы subnet-те орналасқан басқа аккаунттардың ресурстарын көре алмайсыз немесе өзгерте алмайсыз. Және бұл осы shared subnet аясында бірге жұмыс істейтін барлық қатысушылар үшін де солай болады. Бұл тәсіл қосымшалары өзара тығыз байланысты IT-инфрақұрылымдарда өте жақсы қолданыла алады. Осылайша қосымшалар арасындағы байланысты жеңілдетумен қатар, кейбір ортақ ресурстарды үнемдеуге болады, мысалы NAT gateway, VPC Interface Endpoints және т.б.

AWS ішінде жүзеге асыра алатын тағы бір қызықты тәсіл – VPC peering. Бұл әртүрлі VPC-лерді бір-бірімен байланыстыруға мүмкіндік береді. Бұл VPC-лер бір аккаунтта немесе бірнеше әртүрлі аккаунтарда болуы мүмкін. Екі VPC арасында байланыс, яғни peering connection орнату үшін, бірінші – peering connection ресурсын жасау керек, ортасында көріп отырғаныңыздай pcx-id, сонымен қатар сәйкес route tables-терде әр VPC үшін route-терді қосыңыз.

Дәлірек айтсақ, VPC A route table-ына біз route қосуымыз керек, онда destination VPC B CIDR блогы болып табылады және target ретінде біздің peering connection ресурсын көрсетесіз, яғни біз VPC B-ге сол арқылы жетеміз. Ал VPC B route table-ында керісінше, destination VPC A CIDR блогын және target – peering connection ресурсын көрсетесіз.

VPC peering-тің бірнеше шектеулері бар. Бірінші және ең бастысы – IP-адрестер, яғни CIDR блоктар қиылысуы мүмкін емес. Егер олар қабаттасса, сіз peering connection-ді орната алмайсыз. Бұл менің сізге алдыңғы слайдтарда айтқаным, және сіз VPC-ді бір-бірімен қосуды жоспарламасаңыз да, қабаттасуды алдын алу үшін барлық CIDR блоктары бар орталық тізім болуы керек екенін есте ұстауыңыз керек. Тағы бір мәселе, VPC peering-інің өтпелі емес екендігі. Яғни, VPC A және VPC B арасында, содан кейін VPC B және VPC C арасында байланыс орнатсақ, VPC A және VPC C арасындағы байланыс пайда болмайтынын көрсетеді. VPC A-ның VPC C-ге де қосылуын қаласаңыз, бөлек peering connection орнату керек. Сондай-ақ есте сақтау маңызды тағы бір мәселе – екі VPC арасында тек бір peering connection орнатуға болады.

Осы уақытқа дейін біз бұлт ішіндегі желілерді қалай қосуға болатындығы туралы айттық. Егер бізге бұлтқа жергілікті кеңседен немесе жергілікті дата-центрден қосылуымыз керек болса, бұл опция да қолжетімді. Осы мақсатта осының барлығын баптауға мүмкіндік беретін AWS Site-to-Site VPN сервисі бар. Жергілікті кеңседен бұлттағы VPC-ге кіруді қамтамасыз ету үшін сізге қажет ең негізгі қадамдарды қарастырайық. Біз жасауымыз керек ең бірінші қадам – Virtual Private Network Gateway немесе Virtual Gateway құру керек. Сіз экранның ортасынан Virtual Gateway ID көре аласыз. Ол біздің VPC-ге байланады. Әрі қарай, біз Customer Gateway-ді конфигурациялауымыз керек. Customer Gateway жай ғана ресурс емес, бұл AWS-ке сіздің VPN құрылғыңыз туралы ақпарат беретін AWS ресурсы, яғни жабдық. Үшіншіден, біз Private Subnet, route table-ді конфигурациялауымыз керек, мысалы route-ті қосуымыз керек, мұнда destination біздің жергілікті кеңсеміз, яғни жергілікті кеңсеміздің CIDR блогы, ал target – vgw-id болып табылады, яғни бұл Virtual Gateway ID. Осыдан кейін біз екі жүйені бір-бірімен байланыстыру үшін AWS Site-to-Site VPN сервисін пайдалануымыз керек. Бұл жергілікті инфрақұрылымды AWS бұлтындағы инфрақұрылыммен байланыстыру үшін жасалуы керек негізгі қадамдар.

Біздің бұлтты инфрақұрылымымызды жергілікті инфрақұрылыммен байланыстыруға мүмкіндік беретін келесі ұқсас сервис – бұл AWS Direct Connect сервисі. Оның айырмашылығы – алдыңғы AWS Site-to-Site VPN сервисімен біз VPN-қосылымды интернет арқылы баптаймыз. Осылайша, бізде байланыс бар, бірақ деректерді жіберу жылдамдығы мен желілік арна интернет арнасының мөлшерімен анықталады, егер сізде интернет байланысы әлсіз болса, бұл сіздің бұлтқа AWS Site-to-Site VPN сервисі арқылы қосылуыңыз да әлсіз болады дегенді білдіреді. Егер сізге үлкен көлемді деректерді жіберу немесе оны жылдамырақ тасымалдау қажет болса, онда интернетті айналып өтіп, тікелей DX-locations-ке, яғни AWS жаһандық инфрақұрылымға қосылу мүмкіндігі бар. Бұл жағдайда интернетке шығудың қажеті жоқ, біздің жергілікті инфрақұрылымды бұлттағы инфрақұрылыммен тікелей байланысасыз.

Көбінесе VPC-ден қандай да бір AWS сервисін шақыру қажет болады, яғни барлық AWS сервистері VPC ішінде жұмыс істей алмайды, бірақ соған қарамастан бізге VPC-ден AWS ішкі желісінен шықпай-ақ, интернетке қосылмай-ақ қажет AWS сервисіне өтуге мүмкіндік беретін шешім бар. Бұл VPC endpoints, оның екі түрі бар, Interface VPC endpoint немесе AWS Private Link қолдайтын Interface endpoint. Екінші түрі – Gateway endpoint. Сізге қандай сервис қажет екеніне байланысты, endpoint опцияларының қайсысына қолдау көрсетілетінін құжаттамадан қарап, сәйкесінше пайдалануға болады.

VPC жүйесінен Amazon S3 сервисімен жұмыс істеу үшін не баптау керек екенін Amazon S3 сервисінің мысалымен қарастырайық. Ол үшін бізге VPC Endpoint ресурсын құру керек, сонымен қатар subnet-ке арналған route table-де қосымша route тіркеу керек, мұнда destination – Amazon S3 ID, ал target – біздің VPC endpoint ресурсы.

Алдында біз VPC peering туралы, яғни екі бөлек VPC-ді әртүрлі AWS аккаунтарында болса да бір-бірімен байланыстыру екенін айттық. Бұл тәсілдің шектеулерінің бірі транзитивтіліктің жоқтығы болды, яғни сіз бірінші VPC-ді екіншісіне қосасыз, екіншісін үшіншіге қосасыз және бұл біріншіден үшіншіге байланыс бар дегенді білдірмейді. Егер сізге бірінші VPC-ді үшіншіге қосу қажет болса, қосылым пайда болуы үшін бөлек VPC peering connection баптау қажет.

Енді слайдтың сол жағында бірнеше түрлі VPC бар, сонымен қатар VPN қосылымдары бар және олардың барлығы бір-бірімен байланысты болатын жағдайды елестетіп көріңіз. Барлық VPC-лерді бір-бірімен байланыстыру үшін, айта кетейін бұл сіздің бизнес қажеттіліктеріңізге байланысты, сізге VPC peering-інің осы санын жасауыңыз керек. Мұны сақтау өте ыңғайсыз болуы мүмкін және келесі уақыттарда көптеген VPC peering-тер жасалады. Қолданыстағы IT-инфрақұрылымдарда осындай мәселені шешу үшін AWS Transit Gateway сервисін пайдалану ұсынылады. Оның идеясы – бұл Hub and Spoke Model принципі бойынша жұмыс істейді, яғни AWS Transit Gateway орталық хаб ретінде әрекет етеді және Transit Gateway-ге қосылатын барлық VPC-лер AWS Transit Gateway-ге қосылған барлық VPC-ге автоматты түрде қол жеткізе алады. Осылайша, сіз AWS Transit Gateway көмегімен өзіңіздің инфрақұрылымыңыздың көрінісін қалай жеңілдетуге болатындығын оң жақта көре аласыз.

Сонымен біз үшінші бөлімнің соңына жеттік. Өтілген материалға қысқаша тоқталайық. Бірінші VPC компонент – Internet Gateway болып табылады, ол интернетті public subnet-тегі ресурстарымызға байланыстыруға көмектеседі. Содан кейін private subnet ішіндегі инстанстарға қосылу арқылы интернетке кіруге тыйым салу кезінде Private Subnet-тегі инстанстарға интернетке кіруге мүмкіндік беретін NAT Gateway немесе NAT Instance компоненті бар. Әрі қарай, біз VPC Endpoint туралы айттық. Бұндағы идея, барлық сервистер VPC ішінде бола бермейді, және сіздің бизнес талаптарыңызға байланысты, егер сізге AWS сервисін пайдалану қажет болса, интернетке қосылмай-ақ, AWS желісінің ішінде, жергілікті түрде, яғни бұл сервистен VPC-ге қосылған жағдайда, қауіпсіз, тезірек және тұтастай алғанда VPC-ден сізге қажетті сервиске қол жеткізе аласыз. Келесі компонент – VPC peering, бұл VPC-леріңізді, олар әр түрлі AWS аккаунттарында болса да, жұппен байланыстыру қажет болғанда. Келесі компонент – бұл VPC sharing. Мұндағы идея – сіз subnet-теріңізді басқа VPC-лермен, басқа AWS аккаунттарындағы VPC-лермен бөлісе аласыз. Ең бастысы, барлық аккаунттардың бір AWS Organizations-те болуын ескеру қажет. Әрі қарай, біз AWS Site-to-Site VPN жайлы сөйлестік. Идея мынада: бұлтты инфрақұрылымды жергілікті инфрақұрылыммен интернет арқылы арнайы VPN-қосылымы арқылы байланыстыруға болады. Бұл тәсілдің оң және теріс жақтары бар. Басқа нұсқа – AWS Direct Connect. Идеясы – сіз бұлтқа интернет арқылы емес, AWS ғаламдық инфрақұрылымына қосыласыз. Осылайша, сізге интернетке кірудің қажеті жоқ, сонымен бірге сіз кеңірек арнаға ие боласыз. Және бұл опция жергілікті кеңседен бұлтқа немесе керісінше үлкен көлемдегі ақпаратты тасымалдау қажет болса тиімді. Біз атап өткен соңғы сервис – AWS Transit Gateway. Идея мынада: егер сіздің VPC-леріңіз өзара тығыз байланысты болуы керек болса және VPC peering-ті пайдаланған кезде, сіз осы VPC peering-терді көп мөлшерде жасайсыз, бұл жағдайда барлық VPC қосылатын орталық ресурс болатын AWS Transit Gateway шешім бола алады. Осылайша, барлық қосылған VPC-лер осы Transit Gateway-ге қосылған барлық басқа VPC-лерге қол жеткізе алады.

Ал енді төртінші бөлімді бастайық. Бұл бөлімде біз VPC-дегі қауіпсіздік туралы сөйлесеміз. Біз Security Groups-тан бастаймыз. Security Groups сіздің инстанстарыңыз үшін кіріс және шығыс трафикті басқаруға мүмкіндік беретін виртуалды firewall болып табылады. Security Groups инстанс деңгейінде жұмыс істейтінін есте сақтау маңызды. Осылайша, бір subnet-те оның ішіндегі әрбір инстансқа Security Groups-тың арнайы жинағын орнатуға болады. Security Groups кіріс және шығыс трафикті басқаратын ережелер жиынтығы деп түсіну керек. Жаңадан құрылған Security Groups-та сізде Inbound Rules, яғни кіріс трафигі болмайды. Осылайша, үшінші тараптағы хосты жаңадан құрылған Security Group-пен байланысып, инстансқа жете алмайды. Егер біз Outbound Rules, яғни шығыс трафигі туралы айтатын болсақ, онда әдепкі бойынша ол толығымен рұқсат етілген және ешқандай жолмен шектелмейді. Егер шығыс трафикті шектеу қажет болса, онда сіз rule-ді, яғни әдепкі бойынша жасалған ережені жойып, белгілі бір дәрежеде шектелген өзіңіздің ережелеріңізді орната аласыз.

Айта кету керек, Security Groups stateful болып келеді, яғни сіз инстансыңыздан басқа ресурсқа сұраныс жасаған кезде, егер сізде Outbound traffic болса, сұраныс рұқсат етіледі. Содан кейін, сіз сұранысыңызға жауап алған кезде, оған Inbound Rule рұқсат етілгеніне немесе тыйым салынғанына қарамастан, сіз оны аласыз, яғни күйі сақталып қалады. Керісінше, яғни егер сыртқы ресурстардан Inbound Rule рұқсат етілсе, онда ол бұрыннан рұқсат етілгендіктен сырттан келген күйі сақталынады. Біз бұл күйді сақтап, шығыс трафиктің рұқсат етілгеніне қарамастан, ол бәрібір рұқсат етіледі және жауапты кері жібереді.

Security Groups туралы мысалды қарастырайық Мұнда шығыс және кіріс трафиктері үшін бапталған rules бар екенін көруге болады. Егер біз шығыс rules-терге назар аударсақ, біз 80 және 443 порттарына рұқсат етілгенін көреміз, яғни бүкіл интернеттен HTTP мен HTTPS қол жетімділік, яғни интернеттегі кез келген IPv4-адрес біздің инстансқа қосыла алады. Келесі, source ретінде біздің желілік CIDR блок көрсетілген, осылайша біз осы жергілікті желіден 22 порт арқылы, яғни SSH арқылы, Security Group-қа байланысқан инстансқа қосыла аламыз. Егер біз шығыс трафигі туралы айтатын болсақ, біз дерекқорларымыздың Security Group ID-ін destination ретінде көрсете аламыз және 1433 портына рұқсат бере аламыз, осылайша Security Group инстансы 1433 порты арқылы дерекқорларға қосыла алады. Бұл Microsoft SQL Server дерекқорлары үшін әдепкі бойынша орнатылған порт.

VPC-дің қауіпсіздігін қамтамасыз етуге көмектесетін келесі компонент – Network Access Control Lists. Сіз оны Network ACLs деп жиі кездестіресіз. Network ACLs firewall ретінде әрекет етеді, сонымен қатар кіріс және шығыс трафикті subnet деңгейінде басқарады. Бұл міндетті емес, сондықтан оны пайдалану не пайдаланбауды өзіңіз білесіз. Amazon өз тарапынан оны пайдаланып, сіздің барлық VPC-леріңіз үшін кіріс және шығыс rules-терін қайталауды ұсынады.

Біз Network ACLs туралы айтқан кезде, әр subnet Network ACLs-тің біреуіне байланысуы керек. Егер сіз байланысты нақты көрсетпесеңіз, subnet-тер дефолтты Network ACLs-ке байланыстырылады. Мұндағы маңызды мәселе – әр subnet тек бір Network ACL-ге байланыса алады, ал бірнеше subnet-тер бір Network ACL-мен байланыса алады. Бір subnet-ті басқа Network ACL-ге байланыстырған кезде, алдыңғы байланыс үзіледі.

Network ACLs келесідей бейнеленеді. Бұл Network ACLs-тің әдепкі бойынша құрылған мысалы. Әдепкі бойынша, барлық кіріс және шығыс IPv4 traffic рұқсат етілген. Қажет болса, IPv6 traffic-іне де рұқсат етіледі. Бұл жерде айта кету керек, Network ACLs – stateless, яғни күйі сақталмайды. Осылайша, белгілі бір ресурсқа шығыс сұранысты жасаған кезде, егер сізде outbound traffic рұқсат болса, сұраныс орындалады. Сонымен қатар, Network ACLs-тер stateless болғандықтан, жауапты қайтарған кезде, ол кіріс трафигі арқылы рұқсат етілгенін де тексереді, содан кейін сіз жауап аласыз. Егер inbound traffic-те бұл кіріс трафикке рұқсат бермейтіні жазылса, алушыға жауап жетсе де, жауапты көре алмайсыз. Керісіншесі де дұрыс, яғни сыртқы ресурс сізден жауап алуы үшін outbound және inbound traffic-ке рұқсат етілуі керек. Әйтпесе, сіз ресурстан сұрау аласыз, бірақ бұл ресурсқа жауап бере алмайсыз, себебі оны Network ACLs өткізбейді және outbound traffic ішінде сәйкес rule жоқ.

Мұнда сіз толтырылған Network ACLs мысалын көре аласыз. Кастомды Network ACLs жасағанда, барлық кіріс және шығыс трафиктер қабылданбайды. Сондықтан, қажет болса, rules қосу керек, яғни кіріс немесе шығыс трафикке рұқсат беретін ережелер. Мұнда HTTPS және SSH қосылымдары үшін кіріс және шығыс трафикке рұқсат етілгенін көре аласыз. Сонымен қатар, HTTPS арқылы кез келген IPv4-адресін қосуға болады, яғни интернеттен кіру мүмкіндігі бар, ал SSH арқылы тек CIDR блогы 192.0.2.0/24 бар IP-адрестер ғана қосыла алады, яғни мұнда сіз жергілікті желіні көрсете аласыз.

Security groups Network ACLs-тен қалай ерекшеленетіні сізге кішкене түсініксіз болуы мүмкін, сондықтан олардың ерекшеленетін кесте түрінде салыстырайық. Security groups туралы айтатын болсақ, Security groups инстанс деңгейінде жұмыс істейді, ал Network ACLs subnet деңгейінде жұмыс істейді, яғни subnet-ке байланыстырады. Ережелер туралы айтатын болсақ, Security groups рұқсат беретін ережелерді ғана қолдайды, яғни трафикке рұқсат береді. Network ACLs туралы айтатын болсақ, рұқсат беретін ережелер ғана емес, сонымен қатар трафикке тыйым салатын ережелерді орнатуға болады. Security groups – stateful, яғни күйін еске сақтайды. Бұл Outbound rules-ке қарамастан кіруге болатын трафик міндетті түрде шығады дегенді білдіреді. Және керісінше де дұрыс. Егер сізде кірген Outbound traffic болса, онда сізде қандай Inbound rules болса да, ол қайта оралады. Network ACLs – stateless, яғни ол күйді есіне сақтамайды және сұраныс Network ACLs арқылы өтуі үшін оған Inbound rules-те де, Outbound rules-те де рұқсат етілуі керек, әйтпесе сұраныс өтпейді. Төртінші ерекшелік, Security groups трафикке рұқсат беру немесе бермеу туралы шешім қабылдау үшін барлық қол жетімді rule-терді қарап, содан кейін шешім қабылдайды. Ал Network ACLs жоғарыдан төменге дейін барлық rules-терді қарап шығады. Трафикке рұқсат беретін ережелерді тапқан сәтте, қалған ережелер қарастырылмайды және трафикке рұқсат етіледі. Осымен біз төртінші бөлімді аяқтаймыз. Мұнда біз Security groups не екенін және Network ACLs не екенін егжей-тегжейлі талқыладық. Бұл компоненттердің екеуі де біздің VPC-ді қорғауға көмектеседі.

Ал енді бесінші бөлімге келдік. Мұнда біз Amazon Route 53 сервисі туралы сөйлесеміз. Бұл бізге DNS сервисін ұсынатын сервис. Қарапайым тілмен айтқанда, бұл пайдаланушылар www.example.com сайтын ашқан сәтте бұл сервис трафикті тиісті IP-адрестерге, атап айтқанда, біздің инфрақұрылымымыздың IP-адрестеріне, біздің инстанстарымызға бағыттайды дегенді білдіреді. Amazon Route 53 AWS сервистерінің кең ауқымын қолдайды. Бұл тек Amazon EC2 инстанстары ғана емес, сонымен қатар Amazon S3 buckets, Elastic Load Balancing сияқты басқа да сервистер болуы мүмкін. Amazon Route 53 ішінде трафик таргет IP-адрестердің күйіне жауап беру үшін икемді түрде бапталуы мүмкін, яғни бұл сервис қайта бағыттайтын IP-адрестер және сәйкес әрекет етеді, яғни осылайша қабылдаушы тарапынан ақаулар туындаған жағдайда трафикті қайта бағыттауды баптауға болады. Amazon Route 53 арқылы біз домендерді сатып ала алатынымызды да айта кеткен жөн. Amazon Route 53 сервисі парақшасында қандай жоғары деңгейлі домендер қолжетімді екенін көріп, танысуға болады.

Бұл слайдта сіз пайдаланушының сұранысы белгілі бір сайтты ашқан кезде қалай өңделетінін көре аласыз. Пайдаланушы, мысалы, www.example.com сайтының атауын енгізген кезде, бұл сұрау DNS resolver-ге өтеді. Бұл домен атауын және оған қандай IP-адрес қызмет ететінін байламдарын немесе осы IP-адресін қайдан алуға болатыны туралы ақпаратты сақтайтын серверлер. Егер DNS resolver-де бұл ақпарат болмаса, бірақ қайда бару керектігін білсе, ол Amazon Route 53 сервисіне барады, ол өз параметрлерін қарап, сәйкесінше осы DNS resolver-ге IP-адресін қайтарады. DNS resolver содан кейін бұл IP адресті пайдаланушыға қайтарады және бұл пайдаланушыға ашық түрде көрсетіледі. Ол браузер жолағында терген кезде, ол ашық сайтты көреді. Бірақ мұнда екі қадам бар. Біз бірінші қадамды қарастырдық. Екінші қадам, браузер IP-адресті алғаннан кейін, ол осы сұранысты тиісті IP-адреске жібереді және сол жерден жауап алып, оны пайдаланушыға көрсетеді. Іздеу жолағында пайдаланушы басында енгізген домен атауы тұрады.

Amazon Route 53 трафикті бағыттауға және икемді баптауға көмектесетін бірнеше түрлі policy-лерді қолдайды. Біріншісі – Simple routing. Идеясы – сіздің домендік атауыңызға қызмет ететін Amazon Route 53 параметрлерінде бір немесе бірнеше IP-адрестерін көрсету. Сұраныс түскен сәтте бұл сервис кездейсоқ IP-адрестерінің бірін береді, ал пайдаланушы жағындағы браузер пайдаланушының сұрауына жауап алу үшін осы IP-адреске барады.

Келесі жетілдірілген policy – бұл Weighted round robin routing. Оның идеясы – сіз өзіңіздің IP-адрестеріңіздің, трафикке қызмет ететін серверлердің тізімі үшін бірнеше салмақ орната аласыз. Сізде екі сервер бар делік, біреуі қуатты, екіншісінің қуаттылығы төменірек. Бұл жағдайда сіз, мысалы, трафиктің 75%-ын үлкен серверге, ал қалған 25%-ын қуаттылығы төменірегіне жібере аласыз.

Үшінші нұсқа – бұл Latency routing. Бұл алдыңғы екеуіне қарағанда күрделірек келеді. Оның идеясы – бұл нақты IP-адрестерден жауап беру жылдамдығын өлшеп, ең жылдам жауап беретінін таңдайды. Бір мысал келтірейін. Біздің инфрақұрылым екі AWS аймағында орналастырылғанын елестетіп көрейік. Біраз уақыттан кейін және қажетті ақпарат жиналғаннан кейін, Amazon Route 53 пайдаланушыларға осы пайдаланушы үшін жылдамырақ жұмыс істейтін аймақты ұсынады.

Келесі – бұл Geolocation routing. Идея мынада, біз белгілі бір аймақтан пайдаланушыларды осы трафикке қызмет ететін серверлерге немесе IP-адрестерге бағыттай аламыз. Бұл әдетте трафикті қажет тілдегі сайтқа бағыттау үшін қолданылады. Біз, пайдаланушы ретінде, Орталық Азия елінен қосыламыз деп елестетіп көрейік, біз орыс тілін білеміз, бұл жағдайда сайтты сайттың орыс тіліндегі нұсқасына бағыттайтын етіп бапатуға болады. Егер пайдаланушы ағылшын тілінде сөйлейтін елден қосылса, бұл жағдайда пайдаланушы ағылшын тіліндегі трафикке қызмет ететін серверлерге бағытталады.

Келесі policy – бұл Geoproximity routing. Идея мынада, Geolocation routing-тен айырмашылығы біз пайдаланушылардың орналасқан жерін қараған кезде, Geoproximity routing біз инфрақұрылымның орналасқан жерін қарастырамыз және тиісінше, жауапты ұсынамыз.

Келесі – бұл Failover routing. Идея – біз IP-адресті active және passive ретінде жібереміз. Егер трафикті өңдейтін active IP-адрес немесе ресурспен бәрі жақсы болса, барлық трафик сонда бағытталады. Кейбір проблемалар туындаған жағдайда, трафик әр сұрақты жіберіп алмай, өңдеу үшін автоматты түрде резервтік IP-адреске бағытталады. Ал active қайтадан қол жетімді болған кезде, ауыстырып қосқыш іске қосылады және барлық трафик active IP-адрестерге қайта бағытталады.

Ең соңғысы – бұл Multivalue answer routing. Идеясы – барлық алдыңғы жағдайларда біз пайдаланушыға бір нақты IP-адресті қайтаратынбыз. Ал Multivalue answer routing-та біз бірнеше IP-адрестерді жібереміз, содан кейін браузер параметрлерге немесе алгоритмге байланысты қай IP-адреске бару керектігін анықтайды.

Мұнда сіз Amazon Route 53 сервисін пайдаланудың мысалын және оның қандай артықшылықтары бар екенін көре аласыз. Мысалы, біздің инфрақұрылымымыз бірнеше аймақта орналастырылған. Егер бірінші аймаққа жақын аймақтан келген пайдаланушы қосылса, ал Amazon Route 53 тарапынан Latency-based routing орнатылып тұрса, бұл жағдайда пайдаланушы өзіне жақын аймаққа бағытталады және бұл сұраныс сәл алысырақ аймаққа барғаннан гөрі тезірек жауап алады.

Бұл слайдта Failover routing policy бапталынатын AWS парақшасының мысалын көре аласыз. Бұл мысал біздің трафикті бағыттау ережелерін икемді түрде реттеуге болатындығын көрсету үшін келтірілген. Егер біз белгілі бір нақты мысал туралы айтатын болсақ, онда бізде Failure threshold 3 бар екенін көреміз, яғни егер 3 дәйекті сұрау сәтсіз болса, онда бізде ауысу болады және барлық трафик осы трафикке қызмет ететін резервтік IP-адресімізге өтеді. Содан кейін әр 30 секунд немесе 10 секунд сайын, Request interval, біз негізгі болып табылатын active IP-адреске оралу мүмкіндігімізді тексереміз.

Бұл слайдта Failover routing policy-ді баптаудың тағы бір мысалы көрсетілген. Бізде негізгі трафикке қызмет көрсететін инфрақұрылым бар делік және Amazon Route 53-те бұл инфрақұрылым негізгі, яғни Active Route болып табылады. Тиісінше, сұрау қандай да бір динамикалық жауапты қайтару үшін дерекқорға жүгінетін Amazon EC2 инстанстарына жетеді. Ал Amazon Route 53 жағында Failover орнатылған, егер инстанстар қолжетімсіз болса, онда біз үшін Failover policy жұмыс істей бастайды және біз Secondary, яғни Passive IP-адрестерге ауысамыз. Бұл жағдайда трафикті Amazon S3 сервисінде орналасқан статикалық веб-сайтымызға бағыттай аламыз. Осылайша біз барлық пайдаланушыларды дұрыс өңдей алмаймыз, бірақ кем дегенде біз оларды тиісті веб-сайтқа бағыттай аламыз, ол жерде олар азды-көпті түсінікті жауап ала алады және сіз тұтынушыларыңызды жоғалтпайсыз. Amazon EC2 инстанстары сұрауларды сәтті қабылдай алатынын мезгіл-мезгіл тексеріп, Amazon Route 53 жауап алған сәтте ол кері ауысады және барлық трафик Amazon EC2 инстанстарына бағытталады.

Біз осымен бесінші бөлімді аяқтаймыз. Басты тақырыптарды шолып өтейік. Біріншісі, Amazon Route 53 – бұл бізге DNS ұсынатын жоғары деңгейде қол жетімді және ауқымды сервис. Қарапайым сөзбен айтқанда, бұл біздің трафикті домендік атаудан осы сұрауды өңдейтін инстанстарға бағыттайтын сервис. Amazon Route 53 сервисі арқылы біз домен атауларын да сатып ала аламыз. Біз үшін қандай жоғары деңгейлі домендер бар екенін біз осы сервистің ресми парақшасынан көре аламыз. Содан кейін, маңызды нәрсе және нақты AWS емтиханында келуі мүмкін Amazon Route 53 сервисінде қандай Routing policy-лер бар. Шындығында, бұл сұрақтар ең қарапайым сұрақтардың бірі болып табылады. Сізге әрбір Routing policy не екенін түсіну сұраққа дұрыс жауап беру үшін жеткілікті. Сіз біздің Practice test-терімізде осыған ұқсас сұрақтарды кездестіресіз. Осыдан кейін біз Amazon Route 53-ті AWS инфрақұрылымымен пайдаланудың кейбір мысалдарын қарастырдық.

Біз дәрісіміздің соңғы бөліміне жеттік. Бұл бөлімде Amazon CloudFront сервисі туралы айтатын боламыз. Бұл интернет-трафикті өңдейтін барлық дерлік IT-инфрақұрылымдарда қолданылатын негізгі сервистердің бірі. Бұл Content Delivery Network сервисі, яғни CDN. Қарапайым тілмен айтқанда, бұл сіздің барлық медиаңызды пайдаланушыларыңызға жақындататын сервис, осылайша ол сіз үшін жылдамырақ, арзанырақ және жалпы алғанда қауіпсіз жұмыс істейді.

CDN неліктен маңызды екенін және ол бізге пайдаланушы тәжірибесін жақсартуға қалай көмектесетінінің мысалын қарастырайық. Пайдаланушы сұрау жасағанда, ол бастапқы серверге өту және сіздің кейбір деректеріңізді, медиа деректеріңізді серверден сұрау үшін бірнеше қадам, түйін арқылы өтуі мүмкін. Пайдаланушыдан серверлерімізге дейінгі қашықтық өте үлкен болуы мүмкін және бұл сіздің қосымшаңыздың жылдамдығына әсер етеді. Бұл пайдаланушы тәжірибесін нашарлатады. Сондықтан CDN, Content Delivery Network деген тәсіл бар. Идеясы – кейбір түйіндерде медиа деректеріңіз немесе кез келген басқа деректер кэштеледі. Осылайша, пайдаланушы серверлердің түпнұсқалық көздеріне қол жеткізбестен, қажетті контентті ертерек ала алады және мұның бәрі әрбір пайдаланушының әрбір сұранысын өңдеу уақытын жылдамдатады. Осылайша, Content Delivery Network, яғни CDN, біз үшін шешім болып табылады. Content Delivery Network – бұл Amazon-ға қатысы жоқ IT ұғым. Демек, бұл бұрыннан келе жатқан технология, тәсіл. Идея мынада: бұл кейбір контентті (мазмұнды) кэштейтін өзара байланысты серверлер желісі. Контент ретінде HTML парақшалары, CSS парақшалары, JavaScript файлдары, суреттер, видеолар, аудио және т.б. болуы мүмкін, яғни кез келген медиафайл. Сонымен қатар, жетілдірілген SDN динамикалық мазмұнды өңдеп, кэштей алады. Осылайша, пайдаланушыларға бастапқы сервиске жету үшін олардың сұрауын күтудің қажеті жоқ. Шын мәнінде, кэштің ортасында бір жерде сақталатын немесе осы пайдаланушы қабылдайтын дайын жауап бар. Осылайша, пайдаланушы тәжірибесі жақсарады. Егер біз AWS туралы айтатын болсақ, онда CDN, яғни Content Delivery Network болып табылатын арнайы сервис бар, ол Amazon CloudFront деп аталады.

Біз AWS жаһандық инфрақұрылымының не екендігі туралы бұрын айттық. Бұл AWS аймақтары, әрбір аймақта availability zone болады, әрбір availability zone-да дата-центр бар және біз дата-центр деген не екенін де айттық. Сондай-ақ Amazon CloudFront Infrastructure деп аталатын CloudFront-қа жататын параллельді AWS инфрақұрылымы бар. Ол екі компоненттен тұрады. Бұл Edge locations және Regional edge caches. Айырмашылықтары қандай? Edge locations – бұл соңғы пайдаланушыларға ең жақын орналасқан серверлер. Осылайша, ол ең танымал, ең қажетті және жиі сұралатын мазмұнды сақтайды. Regional edge cache – біраз алысырақ орналасқан, ол сәл үлкенірек аймақтың ақпаратын сақтайды. Сондай-ақ онда танымал деректер сақталынады, бірақ Edge locations-та орналасқан деректерге қарағанда танымал емес. Сондай-ақ, бұл Edge locations-те қандай да бір себептермен сәйкес келмейтін деректер болуы мүмкін. Олар Regional Edge cache-інде тасымалданады және сақталады. Regional Edge cache-інен кейін бізде ең алыс қашықтық – бұл бастапқы серверлерге тікелей қол жеткізу. Сонымен, қайталайын, бізде үш түйін бар. Ең алысы – бастапқы серверлерге тікелей қол жеткізу. Мазмұнды Regional Edge cache-інен алатын болсақ, сәл жылдамырақ. Ал Edge locations-тен ең жылдам мазмұнды аламыз.

Amazon CloudFront сервисін пайдаланудың негізгі артықшылықтарын қарастырайық. Біріншісі – бұл сервис жылдам және жаһандық. Келесі – біз соңғы кэштеу серверлерінде қауіпсіздікті баптай аламыз. Біз Amazon CloudFront сервисінің әрекетін AWS Lambda Edge арқылы қосымша бағдарламалай аламыз. Бұл сервис AWS сервистерімен барынша біріктірілген CDN болып табылады. Бұл сонымен қатар интернет пайдаланушыларымызға тікелей серверлерімізден қызмет көрсеткеннен гөрі әлдеқайда тиімді және жылдамырақ шешім.

Amazon CloudFront – ақылы ресурс, бірақ кез келген басқа AWS сервисі сияқты біз тек пайдаланғанымыз үшін төлейміз. Amazon CloudFront сервисінің бөлігі ретінде біз кэш-серверлеріне тасымалдаған деректер көлемі үшін төлейміз. Біз, сондай-ақ, осы кэш-серверлеріне HTTP мен HTTPS сұрауларының саны үшін төлейміз. Invalidation Requests сұраулары үшін қосымша ақы алынады. Бұл сұраныс бойынша барлық кэш-серверлерді деректерімізден тазарту қажет болған кезде. Бұл әдетте қандай да бір қатені тауып, оны тез түзеткенде қажет, сонда біздің пайдаланушыларымыз қосымшаның немесе деректердің жаңартылған нұсқасын алады. Бұл жағдайда біз Invalidation Requests жасаймыз. Тағы бір нұсқа – бізде үлкен жаңарту болған кезде және біз бұл үлкен жаңарту пайдаланушыларымызға жылдамырақ жеткенін қалаймыз. Бұл жағдайда кэшті тазартуымыз керек және сұрауды бастаған барлық пайдаланушылардың бірінші пайдаланушысы CloudFront арқылы сұрау жасайды, ал бұл деректер қол жетімді емес болғандықтан, бастапқы серверлерге сұраныс жасап, еректерді қабылдаған соң, кэштейді. Осы деректерді сұрайтын барлық кейінгі пайдаланушылар енді бастапқы серверлерден емес, біздің Edge серверлерімізден алады.

Тағы бір нәрсе – Dedicated IP custom SSL деп аталатын опцияны баптауға болады. Бұл AWS пайдаланушылары жиі пайдаланбайтын опция, бірақ оның бар екенін есте сақтау керек. Қосымша мәліметтер алу үшін Amazon CloudFront сервисінің ресми құжаттамасын қарауға болады.

Біз дәрістің алтыншы және қорытынды бөлімін аяқтадық. Amazon CloudFront сервисі туралы егжей-тегжейлі талқыладық, сонымен қатар CDN не екенін, не үшін қолданылатынын білдік, сондай-ақ бізге CDN ұсынатын AWS сервисі бар екенін білдік. CloudFront сервисін пайдаланудың артықшылықтары – бұл сервис жаһандық, сіз оны икемді түрде теңшей аласыз, оның күй-өзгерісін бағдарламалай аласыз, сонымен қатар қажетті қауіпсіздік параметрлерін орната аласыз. Бұл AWS сервисі, демек, басқа сервистермен жақсы біріктірілген және сіз өзіңіздің қосымшаларыңыздың пайдаланушыларына үлкен ауқымды ақпарат жіберген кезде өте үнемді шешім болып табылады.

Осымен біз дәрісімізді аяқтаймыз. Бүгінгі өткен маңызды мәліметтерге шолу жасап шығайық. Бұл біз AWS-ке байланысты емес желі негіздерін, бірақ бұл AWS ішіндегі Networking-ке қатысты сервистерді түсінуге көмектесетін ақпаратты қарастырдық. Әрі қарай, біз Amazon VPC сервисімен таныстық, оның қосымша құрамдастарын, соның ішінде қауіпсіздікті қамтамасыз етуге көмектесетін компоненттерін қарастырдық, бұл Security groups және NetworkACLs. Осыдан кейін біз Amazon CloudFront және Amazon Route 53 сервистерімен таныстық, осы сервистерді пайдалану мысалдарын және бұл сервистердің бұлтта инфрақұрылымымызды жақсартуға қалай көмектесетінін қарастырдық.

Бүгінгі өткен сервистер жайлы толығырақ ақпаратты мына сілтемелерден таба аласыз.

Ал енді дәрісіміз аяқталды. Сіз бүгін өзіңіз үшін жаңа ақпарат білдіңіз деп үміттенемін. Келесі белсенділіктерде кездескенше, сау болыңыздар!