Қайырлы күн, құрметті студенттер! Сіздерді кезекті дәрісте көргеніме қуаныштымын. Бүгінгі дәрістің тақырыбы – Compute, яғни бұлтты есептеулер. Бұл тақырып өте маңызды және мен оны сіздермен мүмкіндігінше егжей-тегжейлі талқылағым келеді. Сондықтан біз бұл тақырыпты екі аптаға бөлдік, ал бүгін бірінші бөлімін өтеміз. Сонымен, бастайық.

Бүгінгі дәріс үш бөлімге бөлінген. Бірінші бөлімде біз есептеу ресурстарымен қамтамасыз ететін қолданыстағы AWS сервистері туралы сөйлесеміз. Әрі қарай, екінші және үшінші бөлімдер Amazon EC2, яғни Amazon Elastic Compute Cloud-қа арналған. Amazon EC2 – AWS емтиханына және болашақ жұмысыңызға дайындалу кезінде ең негізгі әрі маңызды сервис, өйткені сіз бұлтта IT-инфрақұрылымын құру кезінде осы сервисті пайдалануыңыз әбден мүмкін.

Бірінші бөлім. Қолданыстағы есептеу сервистеріне шолу. Бұл слайдта бізге есептеу ресурстарымен қамтамасыз ететін сервистердің негізгі тізімі мен біз талқылайтын сервистер белгіленген, олардың кейбіреулеріне егжей-тегжейлі, кейбіреулерін азырақ қарастырамыз. Бірінші және ең негізгісі – Amazon EC2, бұл бізге виртуалды машиналарды ұсынатын сервис. Содан кейін Amazon EC2 Auto Scaling бар, бұл біздің Amazon EC2 инстанстары үшін автоскейлинг орнатуға көмектесетін сервис. Келесі үшеуі – Amazon Elastic Container Registry, содан соң Amazon Elastic Container Service, және үшіншісі – бұл Amazon Elastic Kubernetes Service, яғни қысқаша Amazon ECR, Amazon ECS и Amazon EKS. Олар бір-бірінен қалай ерекшеленеді? Бірінші Amazon ECR – бұл Docker images қоймасы, яғни кескіндерді сақтайтын қойма. Әрі қарай, Amazon ECS – бұл Docker кескіндермен жұмыс істеуге көмектесетін сервис. Келесі – Amazon EKS, ол бізге Kubernetes-пен жұмыс істеуге мүмкіндік береді. Kubernetes – бұл Docker images-ті ыңғайлы жетілдірілген оркестрлеу үшін арналған абстрактілі қолданба, яғни егер сіз Docker қолдансаңыз және қандай да бір сәтте сіз жұмыс істеп жатқан кескіндер саны артса немесе үлкен болса, сіз әдетте Kubernetes, яғни K8S қолдануға ауысасыз. Және тағы бір соған сәйкес сервис – AWS Fargate, ол да контейнерлермен жұмыс істейтін managed сервис болып табылады. Бұндағы идея, AWS әкімшілік жұмыстың көпшілігін өзіне алады, ал біз бұлттағы IT-инфрақұрылымның иелері ретінде контейнерлермен жұмыс істеу үшін негізгі моменттерді, белгілі бір кірістерді ұсынамыз.

Келесі сервистер – AWS Elastic Beanstalk және Amazon LightSail. Олар белгілі бір дәрежеде ұқсас, мен сіздерге сервистерді топтастырылған түрде арнайы ұсынып отырмын, осылайша сіз оны құрылымдалған түрде есте сақтай аласыз. AWS Elastic Beanstalk – бұл жерде сіз кодты, веб-қосымша кодын осы сервиске жүктейсіз және бұл сервис сіздің орныңызға AWS бұлтындағы инфрақұрылымды орнатады, осылайша әкімшілік жұмыстың көп бөлігін өз мойнына алады. Бұл Amazon LightSail-ге де қатысты, оны біз қосымшаларды немесе веб-сайттарды құруға көмектесетін сервис деп айтамыз.

Келесі сервис – AWS Lambda, ол жайлы біз бірнеше рет айтқан болатынбыз. Енді біз оны осы дәрісте толығырақ талқылаймыз. Бұл Serverless Compute Service, бұндағы идея, сіз кодты жазасыз, сол кодты басқару және қызмет көрсету тұрғысынан қалғанының бәрі AWS-те орын алады. Әзірлеушілер үшін бұл тамаша сервис, өйткені ол инфрақұрылымды басқаруға қатысты барлық нәрсені барынша азайтады.

Сондай-ақ, AWS Batch-ты атап өткім келеді, бұл бізге batch jobs-ты, яғни үлкен көлемде немесе топтардағы кейбір жұмыстарды орындауға көмектесетін сервис. Мен сондай-ақ айтқым келетін тағы бір сервис – AWS Outposts, бұл біздің жергілікті инфрақұрылымды AWS интерфейсі арқылы басқаруға мүмкіндік беретін сервис. Осылайша, біз AWS-пен байланысқандай, бірақ шын мәнінде, біздің дата-центрдегі есептеу ресурстарын қолданамыз.

Есептеу сервистерінің белгілі бір бөлігін төрт топқа бөлуге болады. Бірінші топ – IaaS моделі, infrastructure as a service, бойынша ұсынылатын сервистер, біздің жағдайда Amazon EC2. Бұл сервистегі атомдық бірлік, яғни біз жұмыс істейтін ресурс – бұл виртуалды машина. Бұл сервис infrastructure as a service моделі ретінде ұсынылғандықтан, біз осы виртуалды машиналарға толық басшылық жасаймыз. Біз кез-келген операциялық жүйені, кез-келген қосымшаларды орната аламыз. Бізге тек қажеті – осының бәрін жасайтын мамандар. Егер сізде мамандар болса, ІТ-инфрақұрылымға қойылған белгілі бір ерекше талаптар болса, онда Amazon EC2 – бұл біздің таңдауымыз.

Келесі нұсқа – Serverless ресурстары, және AWS Lambda оның айрықша мысалы. Біз кодты жазған кезде, бұл кодты бұлтқа жүктейміз және осы кодты басқаруға және қызмет көрсетуге қатысты барлық нәрсе AWS жағында орындалады. Бұл – сіздің инфрақұрылмыңызды жүргізетін мамандар тапшылығы болған кезде тамаша таңдау.

Үшінші топ контейнерлерге қатысты сервистерден тұрады. Кейінірек контейнерлер туралы толығырақ талқылаймыз. Мұнда контейнерлермен жұмыс істейтін төрт сервис бар. Бұл Amazon ECS, Amazon EKS, Amazon ECR және AWS Fargate.

Төртінші топ – бұл Platform as a Service моделі ретінде ұсынылатын сервис, минималды, атомдық бірлік, ресурс бұл жағдайда бір web application болып табылады және бұл сервис AWS Elastic Beanstalk деп аталады. Идея мынада, біз код жазамыз, осы кодты бұлтқа жүктейміз, содан кейін AWS Elastic Beanstalk қажетті инфрақұрылымды өздігінен орнатады. Әрине, біз осы инфрақұрылымның кейбір шектеулерін, шарттарын немесе сыйымдылығын белгілейтін қосымша параметрлерді ұсынамыз, бірақ бұл инфрақұрылымды орнату және қолдау бойынша барлық жұмыстар AWS Elastic Beanstalk сервисі арқылы жүзеге асырылады, өйткені ол Platform as a Service моделі ретінде ұсынылады.

Есептеу ресурстарына қатысты айтсақ да, AWS сіздің бизнесіңіздің қажеттіліктеріне байланысты сіз үшін өте қолайлы болуы мүмкін бірнеше сервис нұсқаларын ұсынады. Мен мұнда атап өткім келетін жалғыз мәселе, компания дамудың әрқашан әртүрлі кезеңдерінде болады. Осылайша, алдымен, сіз аз жұмыс жүктемесі бар шағын стартап екеніңізді елестетіп көрейік. Бұл жағдайда, AWS Lambda сияқты serverless сервистер сіз үшін өте қолайлы болуы мүмкін. Қызметкерлер штаты болмаған кезде, сізде мұның бәрімен айналысатын толық уақытты жүйелік әкімші жоқ, сонымен қатар сізде IT-шешіміңізге ауыр жүктемелер немесе тұрақты жүктемелер жоқ, бұл жағдайда AWS Lambda-ны пайдалану жеткілікті. Жұмыс жүктемелеріңіз ұлғайған кезде, сіздің қызметкерлеріңіз өседі, содан кейін сізде ІТ-инфрақұрылымға қойылатын ерекше талаптар керек болады, сонда сізге бұлттағы ІТ-ресурстарыңызға көбірек басшылық ету қажет болады. Сіз аз ғана бөлігін жасайтын, ал AWS одан да көп бөлігін орындайтын сервистер ұсынатын модельдерден біртіндеп, сізге IT-ресурстарға толық басшылық ететін сервистерге көшесіз, біздің жағдайда IaaS, яғни Amazon EC2.

Біз сессияның екінші бөлігіне келіп жеттік, енді біз Amazon EC2-мен тереңірек танысамыз. Amazon EC2 – ол толығымен Amazon Elastic Compute Cloud. Бұл бізге виртуалды машиналар ұсынатын сервис. Бұл виртуалды машина жергілікті hosting провайдерінен тапсырыс беретін немесе жергілікті дата-центрде орнатылатын арнайы сервермен ұқсас. Жалғыз айырмашылығы – бұлтты технологиялар жергілікті орнатумен салыстырғанда бұлтты технологиялардың барлық артықшылықтарын ұсынады. Атап айтқанда, сіз есептеу қуатының қажетті мөлшерін бірнеше минут ішінде сұрай аласыз, сізге бұл есептеу қуаты қажет емес кез келген уақытта оны алып тастап, керек кезде кері қайтара аласыз және сіз пайдаланған инстанстар мен инстанс түрлерінің уақыты мен саны үшін ғана ақы алынады. Amazon EC2 сервисін пайдалану арқылы жергілікті инфрақұрылымды бұлтты инфрақұрылыммен салыстырғанда біз қазірдің өзінде кейбір артықшылықтарға ие болып отырмыз. Көптеген IT-жобалардың көпшілігі, ІТ инфрақұрылымға жүктемелері динамикалық болғандықтан, ол біз үшін тиімді. Мысалы, егер бұл киім сататын веб-сайт болса, адамдар әдетте демалыс күндері немесе prime hours деп аталатын сағаттарда, яғни жүктеме шарықтау шегіне жететін уақыттарда сауда жасайды. Бұл уақыт таңғы 9-ға дейін немесе түскі ас, не кешкі 18-19 бен 22 сағаттары аралығында, адамдар жұмыстан босап, біраз сауда жасауға уақыты болады. Бұл жағдайда, сізде жергілікті дата-центрде серверлер болған кезде, барлық жүктемелер бойынша қызмет көрсету үшін және бірде бір клиентті жоғалтпау үшін сізде барынша үнемі жұмыс істеп тұратын серверлердің қандай да бір қоры болуы керек. Өйткені бір минут ішінде жаңа серверді жылдам алу жергілікті хост үшін мүмкін емес тапсырма болып табылады. Серверге тапсырыс беру үшін әдетте бірнеше күн қажет, көп жағдайларда бірнеше апта қажет болады. Сондықтан сізде серверлер қоры болғаны абзал. Тағы бір мәселе – сіз тапсырыс берген серверлерді уақытша пайдалана алмайсыз немесе оларды тез қайтара алмайсыз. Сондықтан, сіз сатып алған кезде, сіз өзіңіздің негізгі шығындарыңызды үлкен мөлшерде жұмсадыңыз деген сөз. Ал егер сізде жүктеме болмаған жағдайда, мысалы таңғы 4 сияқты уақыттарда ешкім сіздің веб-сайтыңызды пайдаланбайды, бұл жағдайда сіздің барлық есептеу қуатыңыз бос тұрады және сол уақыттарда сіз өз ақшаңызды жоғалтасыз. Біз бұлтты сервистармен жұмыс істегенде, скейлинг, автоскейлинг орната аламыз, яғни біздің инфрақұрылымға түсетін жүктемеге байланысты біз Amazon EC2 инстанстарының қажетті мөлшерін аламыз. Бұл бізге ақша үнемдеуге мүмкіндік береді, өйткені жүктеме аз болған кезде біз серверлердің көпшілігін Amazon-ға қайтарамыз және осы сағаттарға ақы төлемейміз, тек осы сауда аз жүретін сағаттарда клиенттерге қызмет көрсетуге қажетті ең аз мөлшерде инстанстарды сақтаймыз. Ал жүктеме ең жоғары деңгейде болған кезде, сайтыңызға адамдар көп кірген кезде, бұлтты провайдерден қажетті инстанс саны сұралады және ол қолжетімді болған минуттарына қарай трафик жіберу мүмкіндігі пайда болады. Бәрі автоматты түрде болады, өйткені бұл бұлттағы тірі организм секілді. Бұл сізде кенеттен жұмыс жүктемесінің он есе ұлғаюы болса да, сізге келген барлық жаңа пайдаланушыларға сәтті қызмет етуге мүмкіндік береді, өйткені сіз жай ғана 10 есе көп инстанстарды сұрай аласыз. Әрине, сіз бұдан пайда табуыңыз мүмкін, бұл сіз үшін өте тиімді болады, ал сауда көп жүретін сағаттар аяқталған кезде сіз оларды қайтарасыз және осылайша сіздің инфрақұрылымыңыз бос тұрған жағдайлар болмайды. Amazon EC2 инстанстарын пайдалану мысалдары жергілікті серверде орнатуға болатын барлық мысалдарды қамтиды, мысалы, қосымшаға, веб-сайтқа, дерекқорға қызмет ететін сервер, мүмкін желі арқылы ойынға қызмет ететін ойын сервері, почта сервері, медиа сервер болуы мүмкін, суреттерді таратқан кезде, аудио немесе видео материалдар болуы мүмкін. Және басқа нұсқа – файл-сервер, файлдарды сол жерге жүктеп, олармен жұмыс істеу мүмкіндігін бергенде, мысалы Dropbox сияқты файлдарды жүктегенде ол қандай да бір proxy-сервер және жергілікті серверге, яғни жергілікті дата-центрде орналасқан серверге қолданылатын барлық мысалдар болуы мүмкін, олар бұлттағы Amazon EC2 инстанстарына да қолданылады. Қысқаша айтқанда, Amazon EC2 – бұл бізді виртуалды машиналармен қамтамасыз ететін сервис, бұл виртуалды машиналар Amazon EC2 инстанстары деп аталады. EC2 – Elastic Compute Cloud деп оқылады.

Сондай-ақ, Amazon EC2 барлық танымал Windows және Linux операциялық жүйелерін, атап айтқанда Windows 8, 12, 16, 19, Red Hat, SUSE, Ubuntu-ды қолдайды және Amazon-ның өзі құрған, өзі қолдап, үздіксіз жетілдіріліп отыратын Amazon Linux бар екенін атап өткім келеді. Қазіргі уақытта екі нұсқасы бар, яғни Amazon Linux 1-нұсқасы және Amazon Linux 2-нұсқасы. Біз 2-нұсқаға көшуді ұсынамыз, өйткені сіз алдыңғы нұсқаға қарағанда бірдей бағаға одан да көп мүмкіндіктер мен артықшылықтарға ие боласыз. Жалпы алғанда, Amazon-ның операциялық жүйесін де пайдалану ұсынылады, себебі ол AWS-пен, сондай-ақ шығатын барлық жаңа элементтермен барынша біріктірілген, олар тек осы операциялық жүйе үшін шығарылуы мүмкін, немесе барлық операциялық жүйелер үшін жаңарту болса, онда ең алдымен ол Amazon Linux ішінде пайда болады.

Amazon EC2 инстансын құру процесіне тоқталайық. Ол 9 түрлі қадамнан тұрады. Ең біріншісі – AMI таңдау. AMI – Amazon Machine Instance, немесе қарапайым тілмен айтқанда, бұл Amazon EC2 инстансының шаблоны, үлгісі. Бұл үлгі операциялық жүйені, барлық мүмкін параметрлерді, орнатылған қосымшаларды және осы Amazon EC2 инстансы аясында жасай алатын барлық операцияларды қамтиды. Яғни, сіз Amazon EC2 инстансын барлық қажетті параметрлермен белгілі бір күйде тоқтатып, бұл күйді жаңа Amazon EC2 инстансына жүктей аласыз. Осылайша, сіз өз уақытыңызды үнемдейсіз, адами факторларды жоясыз және жасалған барлық жаңа инстанстар осы параметрлерді әр Amazon EC2 инстансына нөлден бастап жасағаннан жылдамырақ болады.

АМІ жасаудың үш жолы бар. Біріншісі – өз бетіңізше дайындау, ал екіншісі – AWS Marketplace. Бұл жерде компания өз атынан арнайы параметрлері бар AMI-ді дайындайды және сіз сол жерде осы AMI-ді сатып ала аласыз. Үшінші нұсқа – Community AMI, әлемнің түкпір-түкпірінен келген мамандар өздігінен жасалған AMI-ді дайындап, оларды көпшілікке тегін жариялайтын орын. AWS осы AMI-дың мазмұнын тексермейтіндіктен, AMI-ді Community АМІ-мен пайдалану өз жауапкершілігіңізде болады.

Мен өз тарапымнан AMI-ді Community AMI-мен пайдалануды ұсынбаймын, кейбір жағдайларда, мысалы AWS Marketplace-тен AMI-ді сатып алсаңыз, ол ақталуы мүмкін, бірақ ең жақсы нұсқа – АМІ-ді өз бетіңізше дайындау.

Мұнда диаграммада жаңа AMI құру процесін көре аласыз, яғни бұрыннан бар виртуалды машинаның күйін импорттауға немесе нөлден бастауға болады. Келесі қадамда сіз өзгертілмеген инстанс аласыз, содан кейін сіз барлық қажетті түзетулерді, жаңартуларды, қосымша жаңартуларын, қауіпсіздік патчтарын және басқаларын жасайсыз, нәтижесінде жаңа AMI күйінде сақтайтын өзгертілген инстанс аласыз. AMI белгілі бір аймақ аясында құрылатынын есте сақтаңыз, егер сіздің инфрақұрылымыңыз бірнеше аймақтарда орнатылған болса, сіз басқа аймақта қол жетімді болу үшін АМІ-ді бір аймақтан екіншісіне көшіруіңіз керек. Оның ID-і әр түрлі, бірақ мазмұны бірдей болады.

Келесі қадам – инстанс түрін таңдау. Біз инстанс түрі туралы айтқан кезде, біз осы виртуалды сервердің қуатын айтамыз. Қуат жедел жады мөлшерімен, процессор ядросының санымен, қатқыл дискілерді қосу мүмкіндігі және осы қатқыл дискілердің көлемі туралы ақпарат, сондай-ақ деректерді желі арқылы тарату арнасы бойынша ақпаратпен анықталады. Бұл слайд сервер өлшемінің өзгеруіне байланысты сервер сипаттамалары туралы ақпарат береді. AWS ішіндегі ең кішкентай өлшем – nano, содан кейін micro, small және т.б. Сонымен қатар инстанс атауында префикс жүретінін байқайсыз, әріптерді, біздің жағдайда T және 3 санын көре аласыз. Әріп – бұл инстанстардың тобы, ал сан – буыны. Келесі слайдта инстанстар тобы туралы көбірек сөйлесеміз. Буын туралы айтатын болсақ, бұл жыл сайын артып келе жатқан көрсеткіш; қазіргі уақытта соңғы буын 5 және 6.

Кестедегі өлшемдер туралы айтсақ, сіз инстанстың мөлшері көбейген сайын, процессор ядроларының саны мен жедел жады логикалық түрде артатынын көріп тұрсыз. Мұнда сіз қолданыстағы инстанс топтарының түрі туралы ақпаратты көре аласыз. Сізге әрбір әріпті жаттаудың қажеті жоқ, жай ғана нақты AWS емтиханда келуі ықтимал ең негізгілерін есте сақтасаңыз жеткілікті.

Жалпы General Purprose тобы туралы айтқанда, қара түсті, бұл a, m және t топтарын айтамыз. Есіңізге сақтау керегі – t. Келесі Compute optimized, яғни бірдей өлшем үшін, мысалы, large, compute optimized инстансы general purpose топтың large инстансына қарағанда процессордың ядролары көп санына ие болады, яғни кейбір t3 large және c3 large. Мұнда бұл топ с екенін есте сақтау қажет.

Әрі қарай – r және x болып табылатын Memory optimized, сонымен қатар z бар, бірақ көбінесе ең танымал – r тобы. Мұнда, general purpose салыстырғанда, мысалы, t3 large және бірдей r3 large, бұл жағдайда r3 large тобында t3 large тобымен салыстырғанда кем дегенде екі есе көп жедел жады болады, бірақ процессор ядросының саны бірдей болады.

Сіздің бизнес тапсырмаңызға байланысты төмендегілер сирек қолданылады, бұл кейбір машиналық оқыту модельдерін, жасанды интеллектін үйреткен кезде Accelerated computing тобы. Сондай-ақ, таратылған файлдық жүйелерді орнатуға мүмкіндік беретін инстанс түрлері бар, яғни distributed file systems, Hadoop және т.б. Бұл жағдайда d, h және i топтары бар.

Мен сізге осы видеоны тоқтатып, Amazon EC2 сервисінің Pricing парақшасына баруды ұсынамын. Онда сіз қазіргі уақытта қандай типтегі топтардың бар екенін, қандай өлшемдер бар екенін, қанша тұратынын, сонымен қатар жедел жады мөлшері, неше процессор ядролары қамтамасыз етілгенін және storage және networking мүмкіндіктері қандай екенін көресіз, толық көрініс алу үшін сіз бірнеше түрлі топтарды салыстыра аласыз.

Біз networking-ті қарастырған кезде, networking мүмкіндіктерін айтамыз. Егер сіздің инфрақұрылымыңызда деректердің аз көлемі болса, ол Amazon EC2 инстанстары арасында немесе Amazon EC2 инстанстары мен дерекқор арасында болсын, бұл желідегі ақпарат аса қауіпті емес болғандықтан сізге ұсынылады. Ал егер сіз ақпараттың үлкен көлемімен жұмыс жасап, ол деректер желі арқылы тасымалданса, бұл сіз үшін өте қауіпті болуы мүмкін, сондықтан бұған аса назар аударыңыз. Сондай-ақ, желі үшін оңтайландырылған сатыдағы инстанс түрлерінің арнайы топтары бар екенін атап өткен жөн, яғни оларда жедел жады немесе процессор ядролары онша көп болмауы мүмкін, бірақ ол басқа топтарға қарағанда желі арқылы деректерді беру үшін үлкен арнаны ұсынады.

Келесі қадам, үшінші, бұл – Network settings, онда біз Amazon EC2 инстансы орналасатын барлық ақпаратты көрсетеміз. Атап айтқанда, біз жаңа Amazon EC2 инстансы қай VPC, availability аймағы және қай subnet-те орналасатынын көрсетеміз. Мен мұнда аймақ туралы айтпаймын, өйткені сіз Create Instance немесе Launch Instance батырмасын баспас бұрын, аймақты таңдаулы тұруы керек, өйткені Launch Wizard ішінде сіз аймақты өзгерте алмайсыз. Бұл баса назар аударатын жайт. Егер сіз осы бағдарламаны істеп жатып, онда сіз аймақтардың ешқайсысын көрсетпесеңіз, команда жұмыс істемейді. Мән беретін тағы бір нәрсе – әдепкі VPC-де жасалған барлық инстанстар Public IP-адреске ие болады. Біз бұл динамикалық IP адрес екенін білеміз, бірақ бұл адрес көмегімен сіздің инстансыңыз интернеттен қолжетімді бола алады. Егер әдепкі емес VPC деп айтсақ, жалпыға ортақ IP адресін шығаруға арналған жалауша алынып тасталады және жалпыға ортақ IP адрестер шығарылмайды. Егер сізге жалпыға ортақ IP адресін беру қажет болса, оны VPC немесе subnet құру параметрлерінде нақты көрсетуіңіз керек.

Келесі төртінші қадам – IAM role-ді Amazon EC2 инстансына байланыстыру және бұл қадам міндетті емес болып саналады. Оның идеясы, мысалы, сізде инстанс бар, және оның ішінде AWS сервистерін пайдаланатын және оларға қол жеткізетін белгілі бір код, логика бар деп елестетіп көрейік. Бұл жағдайда оған тиісті қолжетімділік пен құқықтар берілуі керек. AWS IAM role-ін жасау арқылы мұны істеудің ыңғайлы және қауіпсіз әдісін ұсынады, содан кейін осы role-мен instance profile-ін жасайды. Мұның бәрін AWS Management Console арқылы жасаған кезде, IAM role-ін инстансқа байланыстырған кезде, сіз үшін IAM role-мен аттас instance profile ашық түрде жасалады.

Шын мәнінде, instance profile – бұл role-ді қамтитын контейнер және басқа AWS сервистеріне қол жеткізу үшін instance profile-ді инстансқа байланыстыра аласыз. Егер сіз role құрсаңыз және Amazon EC2 инстансына бағдарламалы түрде байланысуға тырыссаңыз, instance profile-ін де құруыңыз керек екенін ұмытпаңыз. Осы слайд біздің role-іміз болған кездегі мысалды суреттейді. Role Amazon S3 bucket-ке қол жеткізуді қамтамасыз етеді және біз бұл role-ді instance profile арқылы инстансқа байланыстырған кезде, инстанс, дәлірек айтқанда, инстанс ішінде жұмыс істейтін код Amazon S3 bucket-теріне қолжетімділікке және қандай да бір түрде сол жерде орналасқан деректермен өзара әрекеттесу мүмкіндігіне ие болады. Сол сияқты, инстансқа кез-келген басқа AWS сервисіне құқық беру процесі осы сияқты өтеді.

Бесінші қадам – User data script, бұл қадам да міндетті емес. Идея, сіз өзіңіздің инстансыңызды жасаған кезде сіз командалар жиынтығын бере аласыз және бұл командалар сіздің инстансыңызды бірінші рет қосқан кезде іске қосылады. Сіз көбінесе инстанс жасалғаннан кейін ғана орындалатын командалардың тізімін бересіз. Олар қолданыстағы қосымшаларды жаңартатын командалар, не болмаса сіздің инстансыңыздың metadata-ларына байланған командалар болуы мүмкін. Мысал ретінде metadata-лар – private IP-адрес, public IP-адрес және басқа ақпаратты айтуға болады.

Егер бізде барлық қажетті бағдарламалары алдын-ала орнатылған, қажетті командалары іске қосылған дайын AMI болса, онда сізде неге User data-ны пайдалануым керек деген сұрақ туындауы мүмкін. Иә, бұл өте дұрыс ой өрісі, және сіз бұның бәрін істеуіңіз керек, яғни AMI үлгісінде орап, сақтауға болатын нәрсенің бәрін жасауыңыз керек. Ал AMI ішіне белгілі бір себептермен орнатуға болмайтын кез-келген әрекетті, сіз User data арқылы жасай аласыз.

User data тек Linux операциялық жүйелері үшін ғана емес, сонымен қатар Windows үшін де қолжетімді. Есіңізде болсын, Windows жүйесінде іске қосылған кезде командалар жиынтығы Windows Command Prompt, яғни CMD немесе Windows PowerShell бағдарламасымен үйлесімді болуы керек. Тағы бір маңызды мәселе – Windows болсын, Linux операциялық жүйесі болсын, сіз іске қосатын барлық командалар admin құқығымен немесе root пайдаланушымен орындалады.

Келесі қадам – алтыншы қадам, Specify storage, яғни мұнда біз қатқыл дискілер туралы ақпарат береміз. Мұнда біз негізгі қатқыл дискіні көрсетеміз, яғни операциялық жүйені орнататын root volume. Сондай-ақ, біз деректерді сақтайтын қосымша қатқыл дискілерді көрсете аламыз. Егер сіз қатқыл дискілерді AMI ішінде баптаған болсаңыз, бұл қадамды өткізуіңізге болады. Қатқыл дискілер туралы айтқанда, біз үш тармақты қарастырамыз: біріншісі – оның өлшемі, келесісі – түрі, ал үшіншісі – осы қатқыл дискілерге арналған шифрлау туралы ақпарат.

Amazon EC2 инстасы үшін сақтау орны туралы айтсақ, біз төрт ықтимал сервисті атап өтеміз: алғашқы екеуі – Amazon Elastic Block Store немесе Amazon EBS және Amazon EC2 Instance Store. Бұл Amazon EC2 сервисінің feature-сі. Бұл сервистердің екеуі де блоктық сақтау болып табылады, сондықтан операциялық жүйені орнатуға және сәйкесінше біздің виртуалды серверді толығымен іске қосуға жарамды. Бірақ бірнеше айырмашылықтары бар, бұрында, Amazon EBS болмаған кезде, тек Amazon EC2 Instance Store сервисі қолданылған. Яғни, бұл дата-центрде физикалық байланған қатқыл дискілер, дәлірек айтқанда біздің жедел жадқа және біздің процессорға қосылған бір блок, бір сервер. Осылайша, бірдеңе дұрыс болмай, бұзылса, сіздің инстансыңыз автоматты түрде ауыстырылса, сол инстанстың қатқыл дискілеріндегі барлық нәрсе сізге қолжетімсіз болады. Осы қолайсыздыққа байланысты балама сервис Amazon Elastic Block Store жасалған. Идея мынада, қатқыл дискілер енді инстанстарға, яғни есептеу қуатына байланбайды, олар бөлек тұрған болып саналады. Осылайша, сіз Amazon EBS арқылы ресурстарды, яғни сіз өзіңіздің инстансыңызға байланыстыратын қатқыл дискілер жасайсыз. Егер инстансыңыз, процессор немесе жедел жады бұзылса, сіз оған тиесілі емес қатқыл дискілерді қоса аласыз, яғни оны басқа инстансқа тіркеп, байланыстыра аласыз және бәрі әрі қарай жұмыс істейді, бастысы сіз ешқандай деректерді жоғалтпайсыз. Бұл Amazon EBS-тің Amazon EC2 Instance Store-дан айырмашылығы.

Сондықтан best practice ретінде жалпы ұсыныс – Amazon EBS-ті мүмкіндігінше пайдалану болып табылады. Кейбір жағдайларда Amazon EC2 Instance Store пайдалану ақталуы мүмкін; бәрі сіздің бизнес мақсатыңызға байланысты және сіз оны егжей-тегжейлі қарастыруыңыз керек.

Келесі нұсқа – Amazon Elastic File System. Бұл да файлдық жүйе, блоктық сақтау. Жалғыз айырмашылығы – сіз оған операциялық жүйені орната алмайсыз, бірақ ол әр түрлі және бірнеше Amazon EC2 инстанстары үшін қол жетімді және кейбір ортақ қатқыл диск болып табылатын оқшау қатқыл диск қажет болған кезде қолданылады. Тағы да бұл сіздің бизнес мақсатыңызға байланысты, сізге кейбір деректерді бір жерде сақтау қажет болуы мүмкін, сондықтан сіздің инстанстарыңыз сол жерге жетіп, сол деректермен бірге жұмыс істей алады.

Бірнеше инстанстармен бірлесіп жұмыс істеудің тағы бір нұсқасы – бізге бұрыннан таныс Amazon S3 сервисі немесе Amazon Simple Storage Service. Бұл сервис блоктық сақтауды емес, объектілік сақтауды қамтамасыз етеді. Объектілермен жұмыс істегенде, біз Amazon S3 сервисін бірнеше инстанстармен жұмыс істеу арқылы бір деректерге қол жеткізу үшін қолдана аламыз.

Мұнда root volume ретінде Amazon EBS-ті пайдаланатын инстанстың және root volume ретінде біздің Ephemeral volume-ді, яғни instance store-ны, пайдаланатын екінші инстанстың мысалдары көрсетілген. Инстанста болатын әрекетке немесе жағдайға байланысты осы екі нұсқаның арасындағы айырмашылықты немесе ұқсастықты қарастырайық. Біріншісі – қайта жүктеу; біз инстансты қайта жүктеген кезде, екі нұсқа үшін де деректерді сақтаймыз және ештеңе жоғалмайды. Тоқтату туралы айтсақ, яғни stop инстанс, Amazon EBS-пен жұмыс істейтін инстансты тоқтата аламыз, тоқтатылған инстанс үшін төлемейміз, тек Amazon EBS ішінде орналасқан деректеріміз үшін төлейміз, ал қымбатырақ тұратын Amazon EC2 үшін төлемейміз. Егер біздің кейбір серверлеріміз жұмыс істемей тұрса және ақшаны жоғалтпау үшін оларды тоқтата тұру керек болса, бұл пайдалы болуы мүмкін. Instance store туралы айтқан кезде, инстансты тоқтату мүмкіндігі қол жетімді емес, сіз оны тоқтата алмайсыз. Сізде қайта жүктеу нұсқасы немесе үшінші нұсқа – серверді жою мүмкіндігі болады. Біз Amazon EBS қатқыл дискілерде бар инстанстарды өшірген кезде, инстанстың өзі жойылады, ал қатқыл дискілер бізге қол жетімді болып қалады. Жою кезінде егер біз жалаушаны белгілеп қойсақ, қатқыл дискілер Amazon EC2 инстансымен бірге жойылады немесе жалауша қоймай біздің қатқыл дискілерді сақтай аламыз. Содан кейін қатқыл дискілерді басқа инстансқа байланыстыруға болады, және онда болған барлық деректер сол күйінше басқа нысанда қол жетімді қалады.

Amazon EC2 Instance Store-мен жұмыс істейтін инстансты өшірген кезде, Ephemeral volume-де болған деректер толығымен операциялық жүйемен бірге жойылады. Сондықтан, қалпына келтіре алмайтын кейбір деректерді немесе маңызды деректерді Amazon EC2 Instance Store сервисінде сақтау ұсынылмайды. Онда сіз кэштеу үшін кейбір деректерді, уақытша немесе кез-келген уақытта қалпына келтіруге болатын басқа деректерді сақтай аласыз.

Келесі қадам – тегтерді қосу мүмкіндігі. Мұнда тегтер кез-келген ресурсқа, кез-келген AWS сервисіне қол жетімді функция екенін атап өткім келеді, сондықтан ол Amazon EC2 инстанстарына тікелей байланысты емес. Тегтер дегеніміз не? Тегтер – бұл кілті және мәні болатын жұптар жиынтығы. Кілттердің бірнеше мысалдарын келтірейін. Ең танымалы name болып табылады, мәнді толтырған кезде, бұл мән көбінесе сол ресурс ID-мен бірге AWS Management Console ішінде көрсетіледі. Көп жағдайда бұл ресурсты анықтауға көмектеседі. Басқа мысал – environment, яғни орта. Бұл әзірлеушілер үшін сынақ ортасы немесе production ортасы болуы мүмкін. Сәйкесінше, key environment үшін мән value жолағында, dev, test, не prod, немесе кез келген басқа мәндер болады.

Тег кілтінің тағы бір нұсқасы – owner, яғни иесі. Бұл мәнде белгілі бір адам, маман немесе сіздің компанияңыздағы белгілі бір басқару бөлімі бола алады. Тегтер идеясы – бұл ресурсты бірегей түрде анықтау және егер бір жағдай болып қалса, жауап табу, табылмаса, онда кем дегенде сізге осы ресурсқа жауапты дұрыс адамдарды, дұрыс бөлімді, ұйымыңыздың құрылымын табуға көмектесетін кейбір ақпаратты ала аласыз. Мұндай ақпаратты немесе адамдарды іздеудің себептері әр түрлі болуы мүмкін. Бұл ресурс ұзақ уақыт бойы жұмыс істеп, ай сайын көп ақша жұмсалуы мүмкін, мысалы, ол мың доллар тұратын Amazon EC2 инстансы десек, ол қомақты қаражат. AWS ішіндегі шығындарды оңтайландыратын адамдар осы ресурсты тауып, бұл ресурстың неге жұмыс істеп тұрғанын, қажет пе, жоқ па және т.б. сұрақтарға жауап алады. Тағы бір нұсқа, қандай да бір сервистердің ішінде ресурстар жиынтығы бар, және бір жерден қате кетсе не бұзылса, оны қалпына келтіре алатын маман табылмауы ықтимал. Бұл жағдайда тегтер бізге сұрақты қайда қою керектігін түсінуге көмектеседі, нақтылау үшін біреуге бұл сервис қажет пе, бұзылған нәрсе қаншалықты маңызды, ол нәрсені қалпына келтіру керек пе, әлде оған уақыт жоғалтпаған жөн бе деген сұрақтарға жауап аламыз, өйткені маманға жұмсаған уақыт та компания үшін ақша.

Барлық тегтер кілт немесе мән болсын, олар case sensitive, яғни регистрге сезімтал. Қарапайым сөзбен айтқанда, сіз сол Name кілтін үлкен әріппен, тек бас әріптермен, сондай-ақ осындай name кілтін кіші әріптермен жаза аласыз. Бұл жағдайда ешқандай қате болмайды, өйткені жүйе бұл кілттерді бөлек кілттер ретінде қабылдайды.

Сегізінші қадам – Security group-тарды баптау. Біз инстансымызға бір немесе бірнеше security group-ты байланыстыра аламыз. Біз бұған дейін security group не екенін талқыладық. Қысқаша айтқанда, бұл – біздің инстансқа дейінгі трафикті фильтрден өткізетін виртуалды firewall. Біз security тобында ережелер жиынтығын көрсетеміз, яғни кіріс немесе шығыс трафикке рұқсат беретін ережелер. Сіз слайдта SSH арқылы қол жетімділікті қамтамасыз ету мысалын көре аласыз, яғни бұл TCP протоколы, 22-порт, және сіз Source көзі ретінде MyIP-ді таңдайсыз. Оны таңдаған кезде ағымдағы адресіңіз көшіріліп, тиісті жолға қойылады. Осыдан кейін сіз SSH арқылы белгілі бір security тобы бекітілген барлық инстанстарға қосылу мүмкіндігіне ие боласыз. Сіздің инстансыңызға сәтті қосылу үшін сізге тиісті SSH кілттері де қажет.

Amazon EC2 инстансын құрудың соңғы қадамы – жаңа key pair таңдау немесе құру. Кey pair – public key пен private key, яғни жалпыға ортақ кілт пен жеке кілт тіркесімі. Жалпыға ортақ кілт сіздің Amazon EC2 инстансыңыздың ішінде сақталады, ал сіз жергілікті сақтау үшін private key жүктейсіз. SSH арқылы қосылғанда, сіз жеке кілтіңізді бересіз, содан кейін private key шынымен Amazon EC2 инстансында сақталған public key-ге қатысты екенін тексеру үшін криптографиялық операция жүреді. Егер бәрі сәйкес келсе, сізге SSH арқылы кіруге рұқсат беріледі.

Мұнда Amazon EC2 болып табылатын AWS аккаунтыңыздағы инстанстар тізімінің мысалын көруге болады. Сол жақ нафигация мәзірінде Instances қойындысы ашылған, және негізгі парақшада барлық инстанстар тізімі көрсетілген. Біз бір инстанстың фильтрден өтіп, таңдалып тұрғанын көреміз. Парақшаның төменгі жағында Description, Status check, Monitoring Tags және басқа қойындыларды көресіз. Бұл қойындылар таңдалған нақты инстанс туралы ақпарат береді.

Сондай-ақ, біз бағдарламалық түрде сіздермен инстанстар құра аламыз немесе AWS бұлтында кез-келген басқа әрекеттерді жасай аламыз. Осы мақсатта AWS CLI немесе Command Line Interface бар, ал екінші нұсқа – AWS Software Development Kit, яғни SDK. Мұнда Amazon EC2 инстансын құратын AWS CLI командасының мысалы келтірілген. Назар аударыңыз, ол бірнеше бөліктен тұрады. Ең бірінші – AWS, біз AWS CLI-ге жүгінетінімізді білдіреді. Содан кейінгі қызмет – Amazon EC2 деп аталады және Amazon EC2 ішінде біз қандай операцияны орындағымыз келетінін көрсетеміз. Біздің жағдайда, бұл run instances операциясы болып табылады. Осыдан кейін, сол жолда біз осы операцияға қатысты параметрлерді көрсетеміз. Мұнда біз Image ID, яғни біздің AMI көрсетілгенін көре аламыз. Әрі қарай, құру керек инстанстардың санын және инстанстардың түрін көрсетіңіз. Key name – жалпыға ортақ SSH кілтінің атауы, security groups және біздің инстанстарымыз жасалатын AWS аймағы туралы ақпарат. Осы команданы іске қосқан кезде, біз жауап ала аламыз. Жауап сәтті не сәтсіз болуы мүмкін. Қате пайда болған жағдайда қате коды туралы ақпарат, сондай-ақ мәтіндік форматтағы қате туралы ақпарат беріледі, бұл сізге қате туралы түсінік береді. Егер біз сәтті жауап алатын болсақ, ол кем дегенде Amazon EC2 инстансының ID-і және осы операция үшін бар және қолжетімді басқа ақпаратты қамтиды.

Бұл слайд кез-келген Amazon EC2 инстансының өмірлік циклі туралы ақпарат береді. Квадрат пішіндері инстанстың күйін көрсетеді, ал көрсеткілердің атауы – біз орындаған әрекет. Басынан бастайық. Бізде AMI бар. Біз instance-ты іске қосқан кезде, инстанс AMI-ден pending күйіне өтеді. Pending күйі AMI дата-центрдің ішіндегі аппараттық құралға орнатылғанын білдіреді. Барлық баптаулар орындалып, инстанс жұмысқа дайын болғанда, яғни ол трафикті қабылдай бастағанда, сіз SSH арқылы қосыла аласыз және ол әдетте пайдалану үшін қолжетімді болады да, ол pending күйінен running күйіне өтеді. Біз білетіндей операциялық жүйе, яғни Amazon EC2 инстансының root volume-і Amazon EBS болуы мүмкін, back-up, яғни Amazon EBS ішінде жасалған қатқыл дискілер немесе Amazon EC2 instance store опциясы пайдаланылады, яғни дата-центр ішіндегі аппараттық құралға тікелей байланған қатқыл дискілермен жұмыс істеген кезде. Екі нұсқа үшін де бізде reboot, яғни серверді қайта жүктеу опциясы бар, бұл жағдайда сервер қайта жүктеледі және екі жағдайда да бар деректер сақталады. Енді, біз Amazon EBS қатқыл дискілерімен жұмыс істеген кезде, бізде біздің инстансты тоқтату мүмкіндігі болады, ол running күйінен, stop әрекетін бастағаннан кейін, stopping күйі арқылы stopped күйіне ауысады. Stopped күйінде біз инстанстарға ақы төлемейміз, ол гибернация күйінде болады және біз тек Amazon EBS үшін төлейміз. Мұнда мен төлемнің негізгі бөлігі қатқыл дискілерге емес, инстанстарға арналғанын еске саламын, сондықтан инстансты тоқтату бізге ақшаны үнемдейді. Біздің инстансымызды қайтадан бастау керек кезде, біз оны stopped күйінен іске қосып, ол pending күйіне өтіп, running күйіне оралады, яғни ол жұмысқа дайын деген сөз. Содан кейін біз ресурстарды, аппараттық құралды terminate, яғни жойып немесе босатып, төлем жасауды тоқтатуға болады. Осылайша, running күйінен, аралық shutting down күйі арқылы, terminated күйіне көшу жүзеге асады. Біздің сервер тоқтатылған кезде, біраз уақыт біздің өшірілген инстансымыз инстанстардың тізімінде terminated күйінде көрсетіліп тұрады. Осы инстансты жою және босату толығымен аяқталғаннан кейін, инстанс тізімнен толығымен жойылады.

Біз Amazon EC2 инстансының өмірлік циклі туралы толығырақ айттық. Сондай-ақ, EC2 инстансының күйіне байланысты жалпыға ортақ IP-адрестермен жұмыс істеудің кейбір нюанстары бар. Бізде жалпыға ортақ IP-адрестің екі нұсқасы бар: біздің инстанстарымызға уақытша берілетін жалпы ортақ, яғни public IP адрес, олар интернет арқылы қол жетімді болады. Екінші нұсқа – static IP адрес, яғни Elastic IP адрес сервисі арқылы берілетін тұрақты IP адрес, егер сізге осы опция қажет болса, оған бөлек тапсырыс беруге болады. Сонымен, public IP адрес туралы айтқанда, жалпыға ортақ IP адрес – бұл қандай да бір pool, яғни аккаунтыңызға арнайы байланыстырылмаған, Amazon үшін қолжетімді IP адрестердің белгілі бір жиынтығы. IP адрестің қаншалықты бос не бос емес екеніне байланысты сізге инстанстарыңызды уақытша пайдалану үшін белгілі бір мән беріледі. Маңызды мәселе, біз инстансты жойып, жаңа инстансты жасаған кезде, біз кездейсоқ жалпыға ортақ IP адресіне ие болатынымыз анық және біз оның мәніне әсер ете алмаймыз. Біз инстансты тоқтатқан кезде, оны басқа AWS аккаунтындағы Amazon EC2 инстансқа тағайындай алатындай етіп, жалпыға ортақ IP адрес AWS-ке қайтарылатынын есте ұстауыңыз керек. Сондықтан, сервер тоқтатылып, қайтадан іске қосылғанда, іске қосу кезінде қол жетімді жаңа жалпыға ортақ IP адрес берілетінін есте сақтаңыз.

Енді статикалық IP адреске оралайық, бұл Elastic IP адрес, сізге статикалық IP адресін беретін арнайы сервис. Әдепкі бойынша, AWS аккаунтыңызда сізге бес IP адрес қолжетімді, ал егер сізге одан да көбірек қажет болса, сәйкес сұрау жасай аласыз. Бұл лимит soft limit болып табылады, сондықтан оны support-ке хабарласу арқылы сіздің қажеттілігіңізге байланысты көбейтуге болады. Сіздің инстансыңыз іске қосылғаннан кейін, сол инстанста метадеректер болады. Метадеректерге жалпыға ортақ IP адресі, жеке IP адресі, осы инстанстың қай аймақта іске қосылғаны, қандай availability zone-ға жататыны, осы инстансқа қандай security топтары байланысты екендігі туралы ақпарат жатады, яғни осы инстансқа қатысты барлық мүмкін ақпарат қолжетімді топталған түрінде AWS Management Console-дің тиісті қойындыларында болады. Егер сіз 169.254.169.254 IP адресі арқылы өтсеңіз, жоғарыда айтылған ақпарат инстанстың өзінен де қолжетімді. Егер біз latest meta-data жолын ұстанатын болсақ, онда біз нақты осы инстанстың метадеректеріне барамыз. Егер біз latest user-data жолымен жүретін болсақ, біз осы инстансты алғаш рет іске қосқан кезде іске қосылатын командалардың тізімін көреміз. Бұл сілтеме тұрақты, және сіз бұл деректерді пайдалана аласыз. Егер сізге белгілі бір инстанс туралы user-data-дан немесе meta-data-дан кейбір ақпарат қажет болса, баптау кезінде осы жолды пайдалануға болады, бұл тиісті кілттері бар JSON құжат, JSON файлындағы бұл жол да тұрақты болып табылады және сіз бұл деректерді инстансыңыздың баптаулары үшін пайдалана аласыз. Сондай-ақ, мен сізге екі URL адресті есте сақтауды ұсынамын, өйткені олар кейде нақты AWS емтиханында келеді.

Біздің Amazon EC2 инстансымыздың мониторингі туралы айтсақ, Amazon CloudWatch сервисі туралы айтамыз, ол метриканы сақтайтын сервис болып табылады және сіз метрикаларыңыздағы өзгерістерді графикалық түрде көрсетіп, әртүрлі әрекет жасай аласыз және өз бетіңізше немесе автоматты түрде тиісті метрикалық мәнге жауап беретіндей баптай аласыз. Метрикалар қалай әрекет ете алады: процессордың ядроларының жүктемесі, қатқыл дискілердегі жүктеме, яғни операцияларды жазу – бөлек, операцияларды оқу – бөлек, сонымен қатар желілік арнаның жүктемесі, яғни Network туралы ақпарат. Мұнда RAM метрикасы, яғни оперативті жады жүктемесі әдепкі бойынша қолжетімсіз екенін атап өткен жөн. Осы метриканы тасымалдай алу үшін бірнеше қосымша қадамдарды орындауыңыз керек, сол кезде бұл метрикалар сізге де қолжетімді болады. Сонымен қатар, CloudWatch аясында бақылаудың екі нұсқасы бар. Бұл Basic Monitoring және Detailed Monitoring. Basic Monitoring опциясында метрикалар әр 5 минут сайын бақылауға жіберіледі және бұл опция тегін және әдеттегідей қолжетімді. Екінші нұсқа – егер сіз өзіңіздің метрикаларыңызды жиі бақылауыңыз керек болса немесе толығырақ ақпарат алуыңыз керек болған жағдайда, Detailed Monitoring сізге метрикаларды минут сайын бақылауға және сақтауға мүмкіндік береді. Айта кететін жайт, Detailed Monitoring бөлек төленеді және сіз төлем мен шығындар туралы ақпаратты тиісті парақшада таба аласыз.

Біздер осымен дәрістің екінші бөлімінің соңына жеттік. Қорытындылап, ең негізгі тармақтарды тағы бір шолып өтейік. Amazon EC2 инстансын құру кезінде біз Linux немесе Windows операциялық жүйелерін таңдай аламыз. Біз белгілі бір AMI арқылы Amazon EC2 инстанстарын құрамыз. AMI – Amazon Machine Instance, ол операциялық жүйеден басқа қосымша параметрлерді, белгілі бір бағдарламаларды орнатуды және белгілі бір командаларды іске қосуды қамтиды, яғни басқа инстанстың снапшоты ретінде әрекет етеді. Amazon EC2 инстансын құрған кезде, біз оның қай VPC-де жасалатынын көрсетуіміз керек. Қауіпсіздік және best practices security тұрғысынан дефолтты VPC-ді пайдаланбай, жаңасын құру ұсынылады. Сізге таңдау үшін процессор ядросының саны, жедел жад өлшемі, сақтау және желі мүмкіндіктері бойынша ең жақсы конфигурацияны табу үшін инсанстардың әртүрлі типтері, инстанс түрлерінің тұтас топтары беріледі. Сіз өзіңіздің инстанстарыңызға кіруді бақылау үшін security топтарын пайдалана аласыз. Security топтары инстанстар деңгейінде жұмыс істейді және виртуалды firewall ретінде әрекет етеді. Инстансты құрған кезде, конфигурацияны аяқтау және инстанс іске қосылғаннан кейін ғана сәтті орындалуы мүмкін командаларды іске қосу үшін user-data деп аталатын командалар тізімін жібере аласыз. Инстанстың операциялық жүйесін Amazon EBS немесе Amazon EC2 Instance Store қатқыл дискілерінде орнатуға болады. Amazon EC2 Instance Store пайдаланатын инстанстарды тоқтату мүмкін емес. Amazon EBS қатқыл дискілерін пайдаланатын инстанстар тоқтатылуы мүмкін. Сонымен қатар, олар тоқтатылған кезде, ол инстанстар үшін төлем жасалынбайды, осылайша қаражатыңызды үнемдейсіз. Инстанстарыңызды бақылау үшін Amazon CloudWatch сервисін қолдана аласыз және онда тиісті метрикаларды жібере аласыз. Сонымен екінші бөлім аяқталды, келесі бөлімге көшейік.

Енді біз дәрістің үшінші бөліміне жеттік. Мұнда біз Amazon EC2 төлемдерін есептеу және оңтайландыру туралы көбірек сөйлесеміз. Amazon EC2 үшін қолжетімді төлем модельдерін қарастырайық. Ең бірінші – on-demand instances. Бұл – біз тек қолданғанымыз үшін ғана төлейтін нұсқа. Біз бірнеше есептеулер жүргізу үшін 2 сағатқа ғана 10 серверге тапсырыс бердік деп елестетіп көрейік, содан кейін бізге бұл серверлер қажет емес делік. Бұл тұрақсыз немесе бір реттік жүктеме есептеулерінің мысалы болып табылады. Осылайша, on-demand айдың соңында бізге 2 сағатты 10 инстансқа көбейтіп, яғни белгілі бір инстанстан 20 сағаттан артық емес уақытқа төлену қажет шотымызды алуға мүмкіндік береді. Яғни біз қолданғанымыз үшін ғана төлем жасаймыз. Мұндағы тағы бір маңызды мәселе – егер біз Amazon Linux немесе Ubuntu операциялық жүйесімен жұмыс істесек, біз секундына төлем жасау мүмкіндігіне ие боламыз. Осылайша, егер біз 1 минут 57 секундқа инстансты іске қоссақ, біз төлемді дәл сол уақыт үшін жасаймыз.

Amazon EC2-дің келесі төлем моделі – Reserved Instances. Бұл төлем моделі жеті күн 24 сағат бойы тұрақты жұмыс жүктемелері үшін өте қолайлы. Бұл жағдайда біз Reserved Instance арқылы есептеу қуатының кейбіріне тапсырыс беріп, AWS-тен жеңілдіктер ала аламыз. Бұл қалай жұмыс істейді? Біз есептеу қуатының белгілі бір мөлшерін бір жылға немесе 3 жылға сатып алуға дайынбыз деп айтамыз. Келісімшартқа қол қойылғаннан кейін, бұл инстанстар бізге жеңілдікпен пайдалану үшін беріледі. Төлемге қатысты кейбір нюанстар бар, атап айтқанда, жеңілдік мөлшерін анықтайтын 3 нұсқа бар. Біз келісімшартқа қол қойған кезде ең төменгі жеңілдікке ие боламыз, бірақ біз ай сайын төлеуді жалғастырамыз. Бұл опция NURI деп аталады, яғни No-Upfront Reserved Instance, біз алдын-ала ештеңе төлемейміз. Содан кейін келесі нұсқа – кезеңнің 50%-ын, бір жыл немесе 3 жылды бірден төлеп, қалған 50%-ын ай сайын төлейміз. Бұл Partial Upfront Reserved Instances немесе PURI деп аталады. Үшінші нұсқа – біз Reserved Instance бойынша максималды жеңілдік алған кезде, AURI, яғни All Upfront Reserved Instances, яғни біз бір жыл не 3 жыл алдын-ала инстанстарымыз үшін төлем жасаймыз. Осылайша, бізге ай сайынғы төлем жасаудың қажеті жоқ. Бұл опция сіз бизнес ретінде өсіп, сізде тұрақты жұмыс жүктемесі бар кезде өте қолайлы. Бұл жағдайда сізде осы тұрақты жүктемелер бар екенін біле отырып, AWS-тен Reserved Instances моделі арқылы жақсы жеңілдік ала аласыз.

Келесі Amazon EC2 төлем моделі – Spot Instances. Бұл өте қызықты модель. Есіңізде болсын, Spot Instances арқылы максималды пайда алуға мүмкіндік береді және жеңілдік 90%-ға дейін жетеді. Бұл уақыт бірлігіне Spot Instance бағалары On-Demand Instances бағасынан 10 есе аз болуы мүмкін дегенді білдіреді. Бұл өте қызықты ұсыныс, оны пайдалану керек. Бірақ мұнда да кейбір нюанстар бар. Spot Instances, мұның бәрі қайдан пайда болды? AWS дата-центрлері өте көп мөлшерде есептеу қуаты бар өте үлкен ғимараттар болғандықтан, кейбір серверлер жүктелмейді, яғни оны ешбір клиент, AWS пайдаланушысы талап етпейді, сондықтан олар жұмыс істемей тұр. Мұнда жағдайда AWS келесі модельді ұсынады. Ол аз болса да табыс табуы үшін, бірақ сонымен бірге босқа ақша жоғалтпайтындай, сұралмаған, басқа төлем модельдері үшін талап етілмейтін инстанстар үшін үлкен жеңілдіктер беруге дайын. Бұл клиенттер үшін де жақсы, өйткені бұл оларға дәл сол аппараттық құралды қолдана отырып, одан да көп үнемдеуге мүмкіндік береді. Мәселе мынада, On-Demand Instances-пен салыстырғанда біз Spot Instance-ке ие бола алмаймыз. On-Demand Instances жағдайында біз инстанстарға тапсырыс береміз, бұл инстанстар біз оларды өзімізден босату туралы шешім қабылдағанға дейін бізге қолжетімді болады, содан кейін оларды босатып, AWS-ке қайта қайтарып, олар үшін төлем жасамау үшін terminated етіледі. Spot Instance жағдайында біз бұл инстанстарды басқа төлем моделімен талап етілгенге дейін қолдана аламыз. Бұл жағдайда сіз осы Spot Instances бұдан былай осы модель бойынша қызмет көрсетіле алмайтындығы туралы хабарлама аласыз және біраз уақыттан кейін ол өшіріледі. Осылайша, егер сізде тұрақты жүктемелер болса, ол Spot Instances моделі бойынша толығымен қамтамасыз етіле алмайды. Spot Instances асинхронды жұмыс жүктемелері үшін өте қолайлы. Сізде кейбір есептеулер бар деп елестетіп көрейік, бұл есептеулер қазір жасалуы мүмкін, сағат түнгі 2-де жасалуы мүмкін, кез келген уақытта үзіліп, сол жерден жалғастырылуы мүмкін, өйткені сіз қандай да бір түрде оны сақтай аласыз. Spot Instances мұндай жүктемелер үшін өте қолайлы. Басқа мүмкін жағдай, масштабтау кезінде, бірнеше топтарды баптау үшін автоскейлингті пайдаланыңыз. Топтың бір бөлігі – On-Demand Instances, ал кейбір инстанстар – Spot Instances. Spot Instances қолжетімді болған кезде, сіз Spot Instances-тармен жұмыс жасайсыз. Егер Spot Instances қол жетімді болмаса, онда AWS барлық Spot Instances-тарын өзіне алады және автоматты түрде автоскейлинг іске қосылады және On-Demand Instances-ті көтереді. Бағасы, әрине, қымбатырақ, бірақ, кем дегенде, осы архитектураның арқасында есептеу қуатын Spot Instanes-қа беру арқылы айтарлықтай қаражат үнемдеуге болады. Бұл қосымша баптауды, маманды қажет етеді және сіз бұл тапсырмаға тиісті есептеумен жүгінуіңіз керек. Егер мұны орнатудың тиімдірек екендігі анықталса, онда оны бизнеске ұсынып, оны жасау керек. Егер артықшылықтар аз болса, онда On-Demand Instances-та қалу ұсынылады.

Spot Instances-та баға белгілеу қалай жұмыс істейді? Ол аукцион секілді ұйымдастырылған, яғни сіз өзіңіздің бағаңызды ұсынасыз, Spot Instanes аясында басқа AWS пайдаланушысы да өз бағасын ұсынады, ал AWS өзінің пайдасын арттыру үшін жоғары бағаны таңдайды, бірақ сонымен бірге бұл баға On-Demand Instances-тан бірнеше есе аз болуы мүмкін. Осылайша, бәрі аймаққа байланысты, мысалы North Virginia AWS-тің ең танымал, ең қаныққан аймағы болып саналады. Ол аймақта Spot Instances пайдаланылуы екіталай. Ал, мысалы, танымал емес аймақтар үшін Spot Instances тамаша нұсқа болуы мүмкін, себебі бұл аймақта өте аз пайдаланушылар отырғандықтан, сіздің тұрақты жүктемелеріңіз көбінесе Spot Instances моделіне сәйкес жұмыс істеп, сіз максималды жеңілдіктерге ие бола аласыз.

Келесі Dedicated Hosts и Dedicated Instances деп аталатын екі төлем моделі Amazon EC2 инстанстың аппараттық құралына байланысты. Олардың қандай айырмашлығы бар? Dedicated Hosts лицензияларды қолдану үшін осы аппараттық құралдың кейбір параметрлеріне қосылуға мүмкіндік береді. Мысалы, Oracle дерекқорының лицензиялары белгілі бір аппараттық құралдағы ядролардың санына байланысты. Бұл жағдайда сіз осы лицензияны қолдану үшін Dedicated Hosts пайдалана аласыз. Dedicated Instances – бұл бұлтта сізге арнайы, басқа AWS пайдаланушыларына жүктеуге мүмкіндік бермейтін, бөлек аппараттық құралдың сервері бөлінеді. Сондықтан сіз оның иесі боласыз, яғни сіз бұл аппараттық құралды толығымен өзіңіз үшін жалға аласыз, онда басқа ешкім болмайды. Бұл жағдайлар кейбір нормативтік талаптар үшін қажет болуы мүмкін. Мысалы, деректердің немесе өңдеудің кейбір ерекшеліктеріне байланысты бұл жүктемелер бөлек аппараттық құралда іске қосылуы керек делік, бұл жағдайда сізге Dedicated Instances-ты пайдалану қажет. Dedicated Instances ерекшеліктеріне байланысты ең қымбат болып саналады, өйткені ол арнайы сізге жеке бөлінеді және бұл өте қымбат қызмет. Сонымен қатар, кейбір нормативтік талаптарға байланысты Dedicated Instances сирек талап етілетінін де атап өткен жөн. Барлық басқа қалыпты жағдайларда Оn-demand instances жеткілікті болады.

Соңғы төлем нұсқасы – Scheduled Reserved Instances болып табылады. Бірден айта кету керек, Scheduled Reserved Instances қазіргі уақытта қолжетімді емес. AWS нақты емтихан аясында сұрақтар жаңартылатынына немесе жиі жаңартылмайтынына байланысты, сізге Amazon EC2 сервисі аясында Scheduled Reserved Instances-ке қатысты сұрақтар келуі мүмкін. Сондықтан, біз қазір бұл мәселені талқылаймыз, бірақ бұл опция қазіргі уақытта AWS ішінде жоқ екенін есте сақтаңыз. Scheduled Reserved Instances дегеніміз не? Бұл күн сайын немесе аптаның белгілі бір күнінде, апта сайын, ай сайын, бір жылға белгілі бір кесте бойынша өзіңіз үшін белгілі бір сағаттарды сақтауға мүмкіндік беретін опция. Осылайша, сіз бір жылға төлейсіз, қай уақытта, қай инстанстар сізге қолжетімді болуы керек екенін көрсетесіз. Және бұл жағдайда олар дәл осы сәтте іске қосылады. Сіз Reserved Instances сияқты, бұл инстанстарды біршама жеңілдікпен төлейсіз.

Келесі екі слайдта біз Amazon EC2 төлем модельдері туралы айтқанымыздың бәрін қорытындылаймыз. On-demand instances туралы айтқанда, бұл мүмкіндігінше икемді нұсқа. Spot instances асинхронды жүктемелер үшін қолайлы нұсқа болып табылады. Reserved Instances – бұл тәулік, апта бойы тұрақты жүктеме болған кезде төлем моделі ретінде жақсы нұсқа. Біз тек бір нақты аппараттық құралды өзіміз қолдануымыз керек жағдайда, кейбір нормативтік талаптарға байланысты арнайы Dedicated hosts қолданамыз.

Use cases туралы айтатын болсақ, On-demand instances болжау қиын қысқа мерзімді динамикалық жүктемелер үшін қолданылады. Бұл жағдайда, On-demand instances тамаша сәйкес келеді. Тағы бір нұсқа – біз әзірлеу немесе тестілеу ортасы үшін On-demand instances қолданамыз, яғни әзірлеу мен тестілеу әрдайым бола бермейді. Сондықтан біз әзірлеу және тестілеу кезеңін аяқтағаннан кейін және бұл инстанстар жұмыс істемей тұрғандықтан, біз оларға ақша жұмсамай өшіре саламыз. Spot instances – бұл бізде асинхронды жүктемелер бар жағдай, онда біз кез-келген уақытта есептеулер мен жүктемелерді тоқтата аламыз және сол жерден қайта жалғастыра аламыз. Ол түнде не таңертең жұмыс істей ме, бұл маңызды емес. Тағы бір мысал – есептеу қуатының үлкен мөлшерінде кейбір есептеулерді жүргізу қажет болған кезде, Spot instances көлемнен біраз пайда әкеледі. Келесі нұсқа – Reserved instances, бізде 24 сағат 7 күн бойы тұрақты жүктемелер болған кезде. Қолданылатын жағдайлардың бірі – компания үлкен болса, жақсы жеңілдіктер алатындығын біле отырып, алдын-ала төлеу мүмкіндігі болған жағдайда Reserved instances пайдалануға болады. Тағы бір нұсқа – біз Disaster recovery орнатқанда, яғни Disaster recovery дегеніміз – біздің инфрақұрылымымыз екі аймақта орнатылған, бірі негізгі, екіншісі резервтік. Бүкіл AWS аймағында қандай да бір мәселе болған жағдайда, әрине бұл өте сирек кездеседі, бірақ біз жүктемелеріміз, инфрақұрылымымыз тұтынушыларға әрқашан 100% қолжетімді болуы керек деп айтамыз, өйткені біраз уақытқа қолжетімсіздікке байланысты клиенттерді жоғалтып, тұтастай алғанда бүкіл компания банкротқа ұшырауы мүмкін. Сондықтан бұл өте маңызды жүктемелер. Бұл жағдайда компания DR, яғни Disaster recovery-ді әзірлеп, қолданады. Егер компания негізгі немесе резервтік аймақтың есептеу қуатының кепілдендірілген мөлшеріне ие болуын қамтамасыз етуі керек болса, оны алдын-ала біле отырып, біз reserved instances арқылы осы есептеу қуатын тағайындап және оны қажет болған кезде іске қоса аламыз. Немесе оны жеңілдікпен алғанымызды ескеріп, үнемі іске қосылып тұрса да болады және біз бұл жағдайда қайта қосқан кезде инстанстардың бізге қолжетімді болатынына 100% сенімді болып, клиенттеріміздің барлық сұраныстарын өңдей аламыз. Соңғы нұсқа – Dedicated Hosts. Оны пайдаланудың бір мысалы, егер біз лицензияны сатып алған болсақ, жағдайға байланысты лицензияларымызды қайтадан пайдалана аламыз, осылайша үнемдей аламыз. Тағы бір use case – белгілі бір нормативтік талаптарды орындау қажет болған кезде, біз басқа адамдардың жүктемелерін тасымалдауға рұқсат етілмеген арнайы аппараттық құралда жұмыс істейміз.

Amazon EC2 сервисінің шығындарын оңтайландыру туралы дәрісімізде айтқанымыздың бәрін қорытындылайық. Ол төрт бағыттан тұрады: Right size, Increase elasticity, Optimal pricing model және Optimize storage choices. Осы бағыттардың әрқайсысын толығырақ қарастырайық. Бірінші бағыт – Right size. Мұндағы идея, біз жұмыс жүктемелерімізге мүмкіндігінше сәйкес келетін инстанстың түрін табуымыз керек. Amazon 60-тан астам инстанстардың түрлері мен өлшемдерін ұсынады, сондықтан нақты біз үшін ыңғайлы біреуін таба аламыз. Бұл процесс бір реттік емес екені анық, бұлтта жұмыс жүктемелері басталмай тұрып, қандай опция сізге тиесілі екенін белгілі бір факторлар арқылы болжап, анықтай алмайсыз. Бұл келесідей болады, сіз мүмкіндігінше арзан және жүктемелеріңізге сәйкес қызмет ететін инстанстың түрін таңдайсыз. Содан кейін, уақыт өте келе, сіз мезгіл-мезгіл көрсеткіштеріңізді, яғни метрикаңызды тексеріп, Amazon CloudWatch сервисіне жібересіз және процессорға, жедел жадқа, сақтау орнына және желіге қандай жүктеме түсетінін көріп, қорытынды жасайсыз.

Сізге қарапайым мысал келтірейін. Сізде CPU 70%-ға және RAM 10%-ға жүктелген instance type бар екенін елестетіп көріңіз. Мұнда біздің қосымшамыз жедел жадқа қарағанда процессордың қуатын көбірек қолданатынын анық көре аламыз. Осылайша, біз кішірек, бірақ CPU болып табылатын топтан, яғни Type-C түрін таңдай аламыз. Бұл жағдайда біз инстанстың мөлшерінде үнемдей аламыз, өйткені ол кішірек болады, бірақ сонымен қатар оның жедел жады да әлдеқайда аз болады. Жүктеме жедел жадын қажет етпейтініне сүйене отырып, бұл инстанстың түрі біз үшін қолайлы. Ол біздің қазіргі жұмыс жүктемелерімізді толығымен басқара алады және алдыңғы нұсқаға қарағанда аз төлейміз.

Екінші бағыт – Increase elasticity. Біз автоскейлингты дұрыс баптауымыз керек. Автоскейлинг – бұл біздің инфрақұрылым тірі абстрактілі ағза секілді, жүктемеге байланысты трафикті өңдейтін инстанстардың саны көбейеді немесе азаяды. Сіз қандай да бір интернет-дүкен екеніңізді елестетіп көрейік және сіздің жүктемелеріңіз көбінесе кешкі 18.00-ден 22.00-ге дейін болады. Бұл жағдайда инстанстарыңыздағы көрсеткіштерді бақылайтын жүйе жүктеменің қазір өсіп жатқанын, серверлердегі жүктеменің өсіп жатқанын анықтайды, сондықтан бұл дұрыс жұмыс істеуі үшін белгілі бір шекті мәннен кейін қосымша серверлерді іске қосу керек. Біз басқа екінші шекті мәннен төмен түскенде, мысалы, егер біздің серверлеріміздегі жүктеме 10%-дан аз болса, онда мұнда үлкен жүктеме жоқ екенін көреміз. Сондықтан біз қосымша инстанстарды өшіріп, оларды AWS-ке қайтара аламыз, сондықтан олар үшін ақы төлеудің қажеті жоқ. Осылайша, біздің инфрақұрылымның мөлшері жүктемеге байланысты азаяды немесе артады. Сондықтан сіз тек сол нақты жүктемелер үшін төлейсіз және сіздің кейбір ресурстарыңыз, инстанстарыңыз бос тұрған кездегі күй болмайды және сіз оны босқа төлемейсіз.

Үшінші бағыт – Оptimal pricing model. Біз бұған дейін Amazon EC2 үшін қолжетімді төлем модельдерін талқыладық. Осылайша, сіздің жұмыс жүктемелеріңізге байланысты сіз үшін ең ыңғайлы және тиімді болатын төлем моделін таңдай аласыз. Сонымен қатар, сіз осы инфрақұрылымды қолдау үшін минималды шығындарды көтересіз.

Келесі нұсқа – Optimize storage choices, яғни сақтау орнын оңтайландыру. Мұнда үш тармақты атап өтуге болады. Бірінші – біз қатқыл дискілердің оңтайлы өлшемін таңдауымыз керек, яғни 20 ГБ негізгі қатқыл дискіні қажет ететін инстанс үшін 500 ГБ қатқыл дискіні құрудың қажеті жоқ. Метрикаңызды жіберетін CloudWatch сервисінің бар екенін есте сақтаңыз және белгілі бір шекке жеткен кезде қатқыл дискілердегі жүктемені, мысалы, 90% қатқыл диск толы дегенді көре аласыз. Бұл жағдайда қатқыл дискілердің өлшемін автоматты түрде ұлғайту үшін кейбір әрекеттерді орындауға болады. Немесе кез келген арналар арқылы, мейлі ол электрондық пошта немесе SMS болсын, қатқыл дискі толғанын хабарлауға болады. Бұл жағдайда сіз өзіңіз байланыса аласыз, не бұл күтпеген жағдай болса, неге бұлай болғанын көре аласыз.

Екінші тармақ – EBS volume түрлері. Келесі дәрістерде біз EBS volume-нің қандай түрлері бар екенін және қай түрді таңдау керектігін талқылаймыз.

Үшінші нұсқа – сізде қандай да бір снапшоттар, яғни қатқыл дискілердің сақтық көшірмелері (бэкап) бар. Бұл кез-келген мәселе туындаған жағдайда оңай қалпына келтіру үшін қажет. Бэкап жасау – қалыпты тәжірибе. Бірақ сонымен бірге бес жылғы сақтық көшірмелерді сақтау әрқашан ұтымды бола бермейді. Сақтық көшірмелерді сіздің бизнес талаптарыңызға байланысты бір апта бұрын, бір ай бұрын, мүмкін бір жыл бұрын сақтау жеткілікті, бірақ біраз уақыт ауысымдарымен, яғни сіз бір жыл бұрын жасалған бір сақтық көшірмені, қайта оралу үшін бір ай бұрын жасалған бір сақтық көшірмені және соңғы жеті күнде жеті сақтық көшірмені сақтайсыз. Бұл – сақтық көшірмелерді сақтаудың ең жақсы нұсқасы. Барлық басқа сақтық көшірмелер жойылуы керек, себебі сіз қолданылмайтын сақтық көшірмелер үшін де төлейсіз, себебі бұл AWS ішінде бөлек жасалған ресурс. Қосымша ресурстар неғұрлым аз болса, соғұрлым аз төлейсіз.

Айта кету керек, бұлттағы шығындарды оңтайландыру процесі бір реттік процесс емес. Бұл бұлтқа тікелей байланысты емес. Бұлт сіздің барлық ресурстарыңызды бір көзқараспен көруге мүмкіндік береді. Бірақ сонымен бірге, мәселе – жүктемелер өзгереді, жаңа ресурстар жасалынады және олар бұлтта да, жергілікті дата-центрде де болады. Сондықтан, мұндағы идея, бұлт жағында максималды көрнекілік ие бола тұра, сіз мұның бәрін өлшей аласыз, оны мезгіл-мезгіл бақылап, оны жақсарту үшін әрекет жасай аласыз. Осы жетілдіруді жасағаннан кейін, осы қадамдардың барлығын циклдік түрде қайталау керек, өйткені инфрақұрылым үнемі өзгеріп отырады, сізде көптеген әзірлемелер болуы мүмкін, және бұл инфрақұрылымның бәрі тірі организм сияқты, ол үнемі динамикалық түрде қозғалыста, онымен үнемі бір нәрсе болып жатады. Сондықтан, өзгерістерді дұрыс өлшеу, бақылау, қадағалау, ағымдағы жағдайды бағалау және кейбір әрекеттерді орындау, осы қадамдардың барлығын мезгіл-мезгіл қайталай отырып, бұлттағы шығындарды оңтайландыру бойынша жұмыс істеудің дұрыс процесі болып табылады. Бұлт бізге барлық қажетті ресурстар мен мүмкіндіктерді ұсынады.

Айта кету керек жайттардың бірі – тегтеу. Тегтеудің арқасында сіз аз топтармен жұмыс істеу үшін ресурстарыңызды логикалық түрде бөле аласыз, өйткені кішігірім топтармен жұмыс істеу оңайырақ. Мысал ретінде, егер сіз өзіңіздің барлық ресурстарыңызды бөлім ID-імен белгілесеңіз, сіз осы немесе басқа проблемалық ресурстардың қай бөлімге тиесілі екенін білесіз. Сіз бұл мәселені шешу үшін қайда бару керектігін білетін боласыз. Сонымен қатар, тегтеуден басқа, шығындарыңызды басқаруға және оңтайландыруға мүмкіндік беретін бірқатар қызықты сервистер бар. Бір Google сұранысымен сіз әрқашан осы ресурстарды таба аласыз, олар туралы оқып, оларды пайдалана аласыз.

Біз сонымен үшінші бөліміміздің соңына, сондай-ақ бүгінгі дәріс сессиямыздың соңына жеттік. Ең маңызды тармақтарға тоқталып өтейік. Біз Amazon EC2 сервисі үшін қолжетімді төлем модельдерін қарастырдық. Бұл On-Demand Instances, Reserved Instances, Spot Instances, Dedicated Instances және Dedicated Host, сондай-ақ қазіргі уақытта қол жетімді емес Scheduled Reserved Instances. Сондай-ақ, егер бізде белгілі бір жүктеме түрі болса, Spot Instances ең тиімді нұсқа екенін білдік. Spot Instances-тің алдын ала 2 минут ішінде өшірілетіні туралы хабарлайтын нотификациясы болуы мүмкін. Тиісінше, сіз өзіңіздің архитектураңызды осы үзілістерге сәйкестендіріп жасауыңыз керек. Сонымен қатар, сіз Spot Instances-ті пайдалана отырып, 90%-ға дейін жеңілдік ала аласыз. Amazon EC2 сервисімен байланысты бұлт шығындарын оңтайландыру үшін ескеру керек төрт тармақты қарастырдық. Бұл Right Sizing, одан соң Increasing Elasticity, яғни автоскейлингті дұрыс пайдалану. Әрі қарай, Optimal Pricing Model, яғни біздің жүктемелерімізге сәйкес келетін модельді таңдау мүмкіндігі. Төртінші – сақтау орнын оңтайландыру, сақтау орындарымен жұмыс істеу кезінде кейбір нюанстар бар, storage сервистері туралы біз келесі дәрістерде толығырақ сөйлесеміз.

Сонымен дәрісіміздің соңына келдік. Біз бүгін Amazon EC2 сервисін мұқият қарастырдық, соның ішінде жаңа Amazon EC2 инстансын жасау үшін қандай қадамдар жасау керектігін толық талдадық. Әрі қарай, біз қандай төлем моделдері бар екенін және Amazon EC2 сервисінің шығындарын қалай оңтайландыруға болатынын қарастырдық.

Осымен сессия аяқталды. Назарларыңызға рахмет. Келесі дәрістерде көрісінкенше, сау болыңыздар!