

Бұлттық есептеулер негізіндегі ақпараттық инфрақұрылымды жобалау

ЖАРИЯЛАНДЫ
26.02.2026

СІЛТЕМЕ
<https://bilimger.kz/187432/>

Очилов Бағлан, Онлашбек Ержігіт, Нұрмахан Ерасыл

Академик Е.А. Бөкетов атындағы Қарағанды Ұлттық зерттеу университеті
студент, 3 курс

Ғылыми жетекші: **Қыдырәлі Дархан Досымбекұлы**

Математика және ақпараттық технологиялар факультеті

Аннотация

Қазіргі заманғы ұйымдардың цифрлық өзгеріс бағытында жылдам дамуы ақпараттық инфрақұрылымға қойылатын талаптардың күрт өсуіне алып келді. Бұлттық есептеулер технологиясы осы талаптарды қанағаттандырудың тиімді жолы ретінде бүкіл әлемде кеңінен қолданылуда. Мақалада бұлттық есептеулер негізіндегі ақпараттық инфрақұрылымды жобалаудың теориялық және тәжірибелік астарлары қарастырылады. Зерттеу барысында бұлттық қызмет көрсету үлгілерінің ерекшеліктері, орналастыру сәулеті, қауіпсіздікті қамтамасыз ету тәсілдері және инфрақұрылымды оңтайландыру стратегиялары талданды. Сонымен қатар мақалада микроқызмет сәулеті, контейнерлеу технологиялары мен желілік инфрақұрылымды жобалау мәселелері егжей-тегжейлі зерделенді. Зерттеу нәтижелері бұлттық инфрақұрылымды тиімді жобалаудың негізгі қағидаттарын анықтауға және оларды тәжірибеде қолданудың ұсынымдарын тұжырымдауға мүмкіндік берді.

Кілт сөздер: бұлттық есептеулер, ақпараттық инфрақұрылым, инфрақұрылым-қызмет ретінде, платформа-қызмет ретінде, бағдарлама-қызмет ретінде, виртуалдандыру, контейнерлеу, микроқызмет, киберқауіпсіздік, масштабтау, деректерді сақтау.

КІРІСПЕ

Ақпараттық технологиялардың қарқынды дамуы кезінде бұлттық есептеулер заманауи кәсіпкерлік үдерістерді ұйымдастырудың маңызды тетігіне айналды. Дәстүрлі сервер инфрақұрылымын ұстап отырудың жоғары шығыны, масштабтаудың күрделілігі және жаңартулардың баяулығы ұйымдарды икемді, үнемді және сенімді шешімдерді іздеуге мәжбүр етті. Бұлттық есептеулер технологиясы нақ осындай шешімді ұсына отырып, соңғы он жыл ішінде жаһандық ақпараттық-технологиялар нарығында үстемдік ететін жаңа бағытқа айналды.

Бүгінде Amazon Web Services, Microsoft Azure, Google Cloud Platform сияқты жетекші бұлттық алаңдар миллиондаған ұйымдарға сан алуан қызметтер ұсынууда. Әлемдік талдау компанияларының деректері бойынша, 2023 жылы жаһандық бұлттық қызметтер нарығының көлемі 591 млрд АҚШ долларынан асты, ал 2026 жылға қарай бұл сан 1 триллион долларды асып кетеді деп болжануда [1, 14 б.]. Бұл сандар бұлттық есептеулердің қазіргі экономикадағы стратегиялық маңызын айқын көрсетеді.

Дегенмен бұлттық инфрақұрылымды дұрыс жобалау — өте күрделі инженерлік міндет. Жобалау кезінде техникалық сәулет шешімдері, қауіпсіздік талаптары, сәйкестік нормалары, шығын оңтайландыру мен өнімділік балансын сақтау мәселелерін бір мезгілде ескеру қажет. Дұрыс жобаланбаған бұлттық инфрақұрылым қауіпсіздік олқылықтарын, артық шығынды және нашар өнімділікті тудыруы мүмкін.

Мақаланың мақсаты — бұлттық есептеулер негізіндегі ақпараттық инфрақұрылымды жобалаудың теориялық негіздерін жүйелендіру, негізгі сәулеттік тәсілдерді салыстырмалы талдау және тәжірибелік ұсынымдар тұжырымдау. Мақалада алдымен бұлттық инфрақұрылымның тұжырымдамалық негіздері мен қызмет үлгілері, одан кейін жобалаудың техникалық астарлары, соңында қауіпсіздік пен басқарылуды қамтамасыз ету мәселелері қарастырылады.

Зерттеу отандық және шетелдік ғылыми еңбектерге, салалық стандарттарға, сондай-ақ бұлттық қызмет берушілердің ресми техникалық құжаттамаларына негізделген. Мақала ақпараттық технологиялар басшылары, бұлттық сәулетшілер және осы саладағы зерттеушілер үшін тәжірибелік пайда тигізуге арналған.

1-ТАРАУ. БҰЛТТЫҚ ЕСЕПТЕУЛЕРДІҢ ТҰЖЫРЫМДАМАЛЫҚ НЕГІЗДЕРІ ЖӘНЕ ҚЫЗМЕТ ҮЛГІЛЕРІ

1.1 Бұлттық есептеулердің анықтамасы, тарихы және негізгі сипаттамалары

Бұлттық есептеулердің ғылыми анықтамасын АҚШ Ұлттық стандарттар және

технологиялар институты 2011 жылы SP 800-145 нормативтік құжатында тұжырымдады. Осы анықтамаға сәйкес, бұлттық есептеу — ортақ баптауға болатын есептеу ресурстарының жиынтығына өзгертіле желі арқылы қол жетімділікті қамтамасыз ететін, ең аз басқару күшімен немесе қызмет берушімен өзара іс-қимыл арқылы жылдам жеткізілетін үлгі [2, 2 б.]. Бұл анықтама бұлттық есептеулердің жай ғана виртуалдандыру немесе хостинг қызметінен бөлек тұратынын айқын бейнелейді.

Ұлттық стандарттар және технологиялар институты бес негізгі сипаттаманы анықтайды: өзіне-өзі қызмет ету мүмкіндігі, кең желілік қол жетімділік, ресурстарды біріктіру, жылдам икемділік және өлшенген қызмет. Осы бес сипаттаманың жиынтығы дәстүрлі ақпараттық-технологиялар инфрақұрылымынан бұлттық есептеуді ажырататын іргелі айырмашылықты анықтайды. Ресурстарды біріктіру принципі бойынша қызмет беруші бірнеше тұтынушыға физикалық ресурстарды ортақ пайдалануға мүмкіндік береді, бұл арқылы жабдықтарды пайдалану тиімділігін едәуір арттырады.

Тарихи тұрғыдан алғанда, бұлттық есептеулердің бастауын 1960-шы жылдарда Джон Маккарти ұсынған есептеу ресурстарын коммуналдық қызмет ретінде жеткізу идеясынан іздестіруге болады. Алайда технологияның нақты дамуы интернеттің жаппай таралуымен, яғни 1990-жылдардың аяғы мен 2000-жылдардың басымен тікелей байланысты. Amazon компаниясы 2006 жылы Elastic Compute Cloud қызметін іске қосқанда, қазіргі заманғы коммерциялық бұлттық есептеулер нарығының іргетасы қаланды. Осыдан кейін Microsoft Azure (2010), Google Cloud Platform (2011) алаңдары іске қосылып, нарық үш ірі ойыншы арасындағы бәсекелестік аренасына айналды.

Бұлттық инфрақұрылымның техникалық негізі виртуалдандыру технологиясында жатыр. Виртуалдандыру физикалық аппараттық ресурстарды логикалық дербес орталарға бөліп, бір физикалық серверде бірнеше виртуалдық машинаны бір мезгілде жұмыс істеуге мүмкіндік береді. VMware компаниясы осы технологияны коммерциялық мақсатта алғаш дамытса, кейіннен ашық бастапқы кодты KVM (ядроға негізделген виртуалдық машина) және Xen гипервизорлары кеңінен тарады. Виртуалдандыру ресурстарды пайдалану тиімділігін 60-80 пайызға дейін арттыруға мүмкіндік береді, бұл бұлттық қызмет берушілердің шаруашылық үлгісінің экономикалық тиімділігін негіздейді [3, 87 б.].

Бұлттық есептеулердің тағы бір маңызды техникалық негізі — бағдарламалық жолмен анықталатын желі және бағдарламалық жолмен анықталатын сақтау тұжырымдамалары. Бағдарламалық жолмен анықталатын желі трафикті басқаруды аппараттық құрылғылардан бөліп, бағдарламалық жолмен орталықтандырылған түрде басқаруға мүмкіндік береді. Бұл бұлттық қызмет берушілерге миллиондаған тұтынушыларға жекелендірілген желі баптауларын ұсынуға мүмкіндік беретін базалық технология болып табылады. Бағдарламалық жолмен анықталатын сақтау деректерді сақтаудың аппараттық жабдықтарынан тәуелсіздігін қамтамасыз ете отырып, жоғары

икемді сақтау жүйелерін ұйымдастыруға жол ашады.

Бұлттық инфрақұрылымның маңызды ерекшелігі — масштабтау мүмкіндігі. Көлденең масштабтау жаңа серверлерді немесе жинақтарды қосу арқылы жүйенің өнімділігін арттырады. Тік масштабтау бар серверлердің ресурстарын — өңдеуіш, жедел жад, сақтау — ұлғайту жолымен жүзеге асырылады. Заманауи бұлттық алаңдар өздігінен масштабтауды қолдайды, яғни жүктеме деңгейіне байланысты инфрақұрылым ресурстарын автоматты түрде ұлғайтып немесе қысқартып отырады. Бұл мүмкіндік шоттық үнемдеуді қамтамасыз ете отырып, қызметтің сенімділігін жоғары деңгейде ұстауға мүмкіндік береді.

Қазақстан Республикасында бұлттық есептеулерді дамыту мемлекеттік деңгейде қолдау табуда. «Цифрлық Қазақстан» мемлекеттік бағдарламасы шеңберінде мемлекеттік органдардың ақпараттық жүйелерін бұлттық инфрақұрылымға көшіру жұмыстары жүргізілуде. Қазақстан Республикасының «Ақпараттандыру туралы» заңы бұлттық қызметтерді пайдалануды реттейтін негізгі нормативтік-құқықтық база болып табылады. Осы аяда отандық ақпараттық-технологиялар компаниялары бұлттық шешімдер нарығын белсенді игеруде [4, 12 б.].

1.2 Бұлттық қызмет үлгілері: инфрақұрылым-қызмет ретінде, платформа-қызмет ретінде және бағдарлама-қызмет ретінде

Бұлттық қызметтің үш негізгі үлгісі бар: инфрақұрылым-қызмет ретінде, платформа-қызмет ретінде және бағдарлама-қызмет ретінде. Бұл үлгілер тұтынушыға берілетін жауапкершілік пен басқару деңгейі бойынша бір-бірінен ерекшеленеді. Қандай үлгіні таңдау ұйымның техникалық мүмкіндіктеріне, реттеушілік талаптарына және стратегиялық мақсаттарына байланысты.

Инфрақұрылым-қызмет ретінде үлгісінде қызмет беруші пайдаланушыға виртуалдандырылған есептеу ресурстарын — виртуалдық машиналар, желілер, сақтау жүйелері — береді. Тұтынушы амалдық жүйені, орта бағдарламалық жасақтаманы және қосымшаларды өзі орнатып, баптайды. Amazon Elastic Compute Cloud, Google Compute Engine, Microsoft Azure Virtual Machines — инфрақұрылым-қызмет ретінде үлгісінің классикалық мысалдары. Бұл үлгі ақпараттық-технологиялар инфрақұрылымын толық бақылауда ұстауды қажет ететін ұйымдарға, сондай-ақ ескі жүйелерді бұлтқа жылжыту стратегиясын таңдаған компанияларға тиімді. Ескі жүйені бұлтқа жылжытудың «көтеріп-жылжыту» тәсілі ең аз қайта жазу шығынын қажет ете отырып, бұлттың икемділік пен масштабтау артықшылықтарынан пайда алуға мүмкіндік береді [5, 43 б.].

Платформа-қызмет ретінде үлгісінде қызмет беруші инфрақұрылым мен амалдық жүйені басқара отырып, тұтынушыға қосымша әзірлеп орналастыруға арналған алаң ұсынады. Тұтынушы тек қосымша кодын жазуға және деректерді басқаруға назар аударады, ал инфрақұрылымдық мәселелермен шұғылданбайды. Google App Engine,

Microsoft Azure App Service, Heroku — платформа-қызмет ретінде алаңдарының кеңінен танымал мысалдары. Бұл үлгі бағдарламашылардың өнімділігін арттырып, бағдарламалық жасақтаманы жеткізу мен пайдалану үдерістерін жеңілдетеді, сондықтан бағдарламалық өнімдерді тез нарыққа шығаруды бірінші орынға қоятын іске қосу кәсіпорындары мен орта кәсіпорындар үшін тиімді.

Бағдарлама-қызмет ретінде үлгісінде тұтынушы дайын бағдарламалық өнімді интернет-браузер немесе ұялы қосымша арқылы пайдаланады. Инфрақұрылым, амалдық жүйе, орта бағдарламалық жасақтама және қосымшаның өзі — барлығын қызмет беруші басқарады. Microsoft 365, Google Workspace, Salesforce CRM, Zoom — бағдарлама-қызмет ретінде үлгісінің ең кеңінен тараған мысалдары. Бұл үлгі тұтынушының ақпараттық-технологиялар ресурстарына тәуелділігін азайтады және орнату, жаңарту мәселелерінен босатады. Алайда бағдарлама-қызмет ретінде үлгісінде деректерді бақылау деңгейі ең төмен болады, бұл деректер жеке өміріне байланысты сезімтал секторлардағы ұйымдарда — медицина, мемлекеттік сектор, қаржы — осы үлгінің қолданысына шектеу қою мәселесін туғызады.

Аталған үш үлгіден бөлек, функция-қызмет ретінде немесе сервердің жоқтығы (serverless) деп аталатын жаңа тәсіл кеңінен таралуда. Бұл үлгіде тұтынушы жеке функцияларды немесе микроқызметтерді орналастырады, ал қызмет беруші осы функциялар шақырылған кезде ғана серверлерді іске қосады. AWS Lambda, Google Cloud Functions, Microsoft Azure Functions — осы үлгінің негізгі алаңдары. Функция-қызмет ретінде үлгісі оқиғаға бағытталған сәулеттер үшін ерекше тиімді, өйткені сервер жұмыс істемеген уақытта ешқандай ақы алынбайды. Дегенмен бұл үлгінің «суық іске қосу» мәселесі — функция алғаш шақырылғанда болатын кешігу — кейбір жоғары өнімділік талаптарын қоятын жүйелерде шектеуші фактор болуы мүмкін [6, 156 б.].

Орналастыру үлгісі бойынша бұлттық инфрақұрылым төрт түрге бөлінеді: жалпы бұлт, жеке бұлт, будандастырылған бұлт және көп бұлт. Жалпы бұлтта ресурстар бірнеше тұтынушы арасында бөліседі, бұл шығынды азайтады. Жеке бұлтта ресурстар бір ұйымға арналған, деректер жеке өміріне жоғары талаптар қоятын ұйымдар үшін қолайлы. Будандастырылған бұлт жеке және жалпы бұлтты біріктіреді, ұйымдарға деректерінің бір бөлігін ішкі серверде, ал басқа бөлігін жалпы бұлтта сақтауға мүмкіндік береді. Көп бұлт стратегиясы бірнеше бұлт қызмет берушілерінің қызметтерін бір мезгілде пайдалануды білдіреді. Бір қызмет берушіге тәуелділік тәуекелін азайту және нарықтық бәсекелестікті пайдалану тұрғысынан көп бұлт стратегиясы барған сайын танымал болуда.

2-ТАРАУ. БҰЛТТЫҚ ИНФРАҚҰРЫЛЫМДЫ ЖОБАЛАУДЫҢ

ТЕХНИКАЛЫҚ АСТАРЛАРЫ

2.1 Сәулеттік тәсілдер: монолиттік жүйе, микроқызмет және контейнерлеу

Бұлттық инфрақұрылымды жобалауда ең маңызды сәулеттік шешімдердің бірі — қосымша сәулетін таңдау. Тарихи тұрғыдан алғанда дәстүрлі қосымшалар монолиттік сәулетте жазылды: барлық қызметтік мүмкіндіктер бір тұтас кодтық базада біріктірілген. Монолиттік сәулет салыстырмалы түрде қарапайым жобалауды, тестілеуді және орналастыруды қамтамасыз етеді. Алайда жүйе өскен сайын монолиттің кемшіліктері анықталады: кодтық базаның күрделенуі, аз ғана өзгерісті енгізу үшін бүкіл жүйені қайта орналастыру қажеттілігі, масштабтаудың тиімсіздігі. Тек бір құрамдасқа ғана жоғары жүктеме болса да, бүкіл жүйені масштабтауға тура келеді, бұл ресурстарды ысырап қылады.

Микроқызмет сәулеті осы мәселені шешу жолы ретінде пайда болды. Микроқызметте қосымша кішігірім, дербес орналастырылатын қызметтерге бөлінеді. Әр қызмет өзінің нақты кәсіпкерлік қызметін орындайды, өзінің деректер қорымен жұмыс жасайды және басқа қызметтермен бағдарламалық байланыс нүктесі арқылы байланысады. Netflix, Amazon, Uber сияқты технологиялық алыптар микроқызмет сәулетіне ертерек көшіп, дербес масштабтаудың, жылдамдықтың және сенімділіктің айтарлықтай артқанын байқады [7, 34 б.]. Дегенмен микроқызмет сәулеті өзінің күрделіліктерін де ала келеді: қызметаралық байланысты басқару, деректердің тұтастығын қамтамасыз ету, бөлінген транзакцияларды өңдеу — барлығы монолитпен салыстырғанда күрделірек мәселелер.

Контейнерлеу технологиясы, ең алдымен Docker алаңы, микроқызмет сәулетінің кеңінен таралуына серпін берді. Контейнер — қосымша кодын, оны орындауға қажетті барлық тәуелділіктерді — кітапханалар, баптау файлдары — бір тұтас орауышқа жинайтын жеңіл виртуалдандыру тетігі. Контейнер виртуалдық машинадан айырмашылығы — амалдық жүйенің ядросын үй жүйемен бөліседі, сондықтан ресурс шығыны мен іске қосу уақыты айтарлықтай аз. 2013 жылы Docker ашық бастапқы кодта шыққаннан бері контейнерлеу ақпараттық-технологиялар саласының стандартты тәжірибесіне айналды. Бүгінде Docker Hub-та миллиондаған дайын контейнер суреттері орналасқан, бұл бағдарламашылардың жұмыс өнімділігін айтарлықтай арттырады.

Контейнерлердің ауқымды орналастыруын басқару үшін үйлестіру алаңдары қажет. Kubernetes — Google компаниясы 2014 жылы ашық бастапқы кодта жариялаған контейнер үйлестіру жүйесі — іс жүзіндегі стандартқа айналды. Kubernetes контейнерлерді автоматты орналастыруды, масштабтауды, жаңартуды және жұмысқа қабілеттілікті өздігінен қалпына келтіруді қамтамасыз етеді. Kubernetes топшасы басқару тегі мен жұмысшы түйіндерден тұрады. Басқару тегі прикладтық бағдарламалау интерфейсі серверін, кілт-мән деректер қорын, жоспарлаушы мен

бақылаушыларды қамтиды. Жұмысшы түйіндер болса, Pod деп аталатын контейнер топтарын іске асырады. Kubernetes-ті қолдану инфрақұрылым операцияларын автоматтандырып, адам қателіктерінен туындайтын инциденттер санын едәуір азайтады.

Қызметтер торы технологиясы микроқызмет ортасында қызметаралық байланысты басқаруды жеңілдетеді. Istio немесе Linkerd сияқты қызметтер торы шешімдері трафик шифрлауды, жүктеме теңгерімдеуді, деректерді жинауды және қызметаралық кешігуді азайтуды инфрақұрылым деңгейінде қамтамасыз етеді, яғни бұл мүмкіндіктерді қосымша кодына енгізудің қажеті жоқ. Бұл тәсіл «бөлу және ережені орталықтандыру» қағидатын жүзеге асырып, бағдарламашылардың тек кәсіпкерлік логикаға назар аударуына мүмкіндік береді. GitOps тәсілі болса, инфрақұрылым баптауын нұсқалық бақылауда жүргізіп, үздіксіз жеткізу желісі арқылы автоматты орналастыруды жүзеге асырады, нәтижесінде инфрақұрылым-код ретінде қағидаты толық іске асырылады [8, 201 б.].

2.2 Желілік инфрақұрылым, сақтау жүйелері және өнімділікті оңтайландыру

Бұлттық инфрақұрылымның желілік сәулеті жобалаудың маңызды бөлігін құрайды. Виртуалдық жеке бұлт — бұлттық қызмет берушінің физикалық желісінде логикалық тұрғыдан оқшауланған желілік кеңістік. Виртуалдық жеке бұлт ішінде ұйым өз желілік топологиясын толық бақылайды: Интернет-протокол мекенжай кеңістігін анықтайды, ішкі желілерді жасайды, бағыттау кестелерін баптайды. Ішкі желілер жалпы қол жетімді — интернетке тікелей қол жетімді — және жеке — тек виртуалдық жеке бұлт ішінен қол жетімді — болып бөлінеді. Деректер қорлары, микроқызмет артқы жақтары, аутентификация серверлері барлығы әдетте жеке ішкі желіде орналастырылады.

Желілік қауіпсіздікті қамтамасыз ету үшін қауіпсіздік тобы және желілік қол жетімділікті бақылау тізімі тетіктері қолданылады. Қауіпсіздік тобы қызмет деңгейіндегі күйлік желіаралық қалқан ретінде жұмыс жасайды. Ол нақты виртуалдық машина инстанстарына немесе Pod-тарға қолданылады және кіріс-шығыс трафигін порт, хаттама және Интернет-протокол мекенжай бойынша реттейді. Желілік қол жетімділікті бақылау тізімі болса, ішкі желі деңгейіндегі күйсіз желіаралық қалқан ретінде жұмыс жасайды. Осы екі тетіктің бірге қолданылуы қорғанысты тереңдету қағидатын іске асырады.

Желілік трафикті оңтайландыруда мазмұнды жеткізу желісі маңызды рөл атқарады. Мазмұнды жеткізу желісі статикалық мазмұнды — суреттер, бейне, стиль кестесі файлдар — пайдаланушыға географиялық жақын серверлерде кэштеу арқылы жүктелу уақытын айтарлықтай қысқартады. CloudFront, Fastly, Cloudflare — кеңінен таралған мазмұнды жеткізу желісі алаңдары. Жүктеме теңгерігіші кіріс трафигін бірнеше сервер немесе Pod арасында бөліп, жүйенің жоғары қол жетімділігін қамтамасыз етеді.

Қосымша деңгейінің жүктеме теңгерігіші Гипермәтінді тасымалдау хаттамасы трафигін жетілдірілген бағыттау ережелері бойынша бөлсе, Желі деңгейінің жүктеме теңгерігіші TCP/UDP трафигін миллиондаған сұрау деңгейінде өте аз кешігіспен өңдейді.

Деректерді сақтау жүйелерін жобалауда мәліметтердің табиғаты мен қол жетімділік талаптарын ескеру маңызды. Блоктық сақтау — Elastic Block Store, Azure Managed Disks — амалдық жүйе томдары мен реляциялық деректер қорлары үшін қолданылады, жоғары кіріс-шығыс амалдар жиілігін қамтамасыз етеді. Нысандық сақтау — Amazon S3, Google Cloud Storage — байтты мазмұнды шектелмеген ауқымда, арзан бағаға сақтауға мүмкіндік береді; медиа файлдар, резервтік көшірмелер, журнал файлдары үшін тиімді. Файлдық сақтау — Amazon Elastic File System, Azure Files — бірнеше инстанс немесе Pod-қа бір мезгілде қол жетімді болуы тиіс ортақ файл жүйесі қажет жағдайларда қолданылады [9, 165 б.].

Деректер қорын таңдауда реляциялық немесе реляциялық емес үлгілердің арасындағы мақсатты таңдау жасалады. Реляциялық деректер қорлары — PostgreSQL, MySQL, Amazon Aurora — атомдылық-дәйектілік-оқшаулану-тұрақтылық транзакцияларын, күрделі сұраулар тілі сұраулары мен деректер тұтастығын талап ететін жүйелерде жоғары тиімділік береді. Реляциялық емес деректер қорлары жоғары жазу жылдамдығы, көлденең масштабтау және икемді схема талаптарын қанағаттандырады. DynamoDB — шексіз масштабтауды қамтамасыз ететін кілт-мән қоры; MongoDB — икемді JSON-тәрізді нысандарды сақтайтын нысандық қор; Cassandra — миллиондаған жазбаны секундына жазу-оқу керек болатын нәрселер Интернеті және телеметрия жүйелерінде тиімді бағандық деректер қоры.

Кэштеу тетіктері деректер қорының жүктемесін азайтып, жауап беру уақытын қысқарту үшін маңызды. Redis — жедел жадтағы кілт-мән қоры ретінде сеанс деректерін, жиі оқылатын мазмұнды және уақытша нәтижелерді кэштеу үшін кеңінен қолданылады. Memcached қарапайым жоғары өнімді кэштеу үшін балама шешім. Деректер базасы алдындағы кэш деңгейі дұрыс баптауланған жағдайда деректер қорының сұрауларының 80-90 пайызын тосып алады, нәтижесінде жүйенің жалпы өнімділігі бірнеше есе артады. Кэш жанынан қол жеткізу, оқу барысында кэштеу, жазу барысында кэштеу, кешіктіріп жазу — кэш стратегияларының негізгі түрлері болып табылады, олардың әрқайсысы нақты пайдалану жағдайының деректерді жаңарту жиілігі мен дәйектілік талаптарына қарай таңдалады.

Инфрақұрылым-код ретінде тәсілі заманауи бұлттық операцияларда орталық рөл атқарады. HashiCorp Terraform — көп бұлттық инфрақұрылым-код ретінде құралы ретінде бұлттық ресурстарды декларативтік HCL тілінде сипаттауға мүмкіндік береді. Terraform жоспарлау, қолдану және жою кезеңдеріне ие. Ansible болса, серверлер мен инфрақұрылым құрамдастарының баптауын автоматтандыру үшін кеңінен қолданылатын баптауды басқару құралы. AWS CloudFormation мен Azure Resource

Manager қызмет берушілерге тән жергілікті инфрақұрылым-код ретінде шешімдер. Инфрақұрылым-код ретінде тәсілінің артықшылығы — инфрақұрылым баптауының нұсқалық бақылауда болуы, топтар арасындағы ынтымақтастықтың жеңілдеуі және қайталанатын, болжамды орналастыруларды жүзеге асыру мүмкіндігі. Мұндай жобалау тәсілі «Ешқашан қолмен өндірістік ортаны өзгертпе — тек код арқылы өзгерт» ережесін жүзеге асырады [10, 89 б.].

Үздіксіз кіріктіру және үздіксіз жеткізу тәжірибесі бұлттық инфрақұрылымды жобалаудың ажырамас бөлігіне айналды. Үздіксіз кіріктіру кезінде әрбір кодтық өзгерту автоматты түрде жинақталады, тексерілуден өтеді және тестілеу ортасына орналастырылады. Үздіксіз жеткізу үдерісі бекітілген кодтық өзгертулерді автоматты түрде өндірістік ортаға жеткізеді. Jenkins, GitLab CI/CD, GitHub Actions, AWS CodePipeline — осы тәжірибені жүзеге асыратын кеңінен танымал құралдар. Бағдарламалық жасақтаманы жасау мен пайдалану үдерістерін біріктіру тәжірибесі — нәтижесінде жобалаудан бастап өндірістік ортаға дейінгі уақыт айтарлықтай қысқарады, деплойменттер жиілейді және инциденттерден қалпына келу жылдамдайды.

3-ТАРАУ. БҰЛТТЫҚ ИНФРАҚҰРЫЛЫМДАҒЫ ҚАУІПСІЗДІК ЖӘНЕ БАСҚАРЫЛУ

3.1 Бұлттық қауіпсіздік сәулеті және нөлдік сенім үлгісі

Бұлттық инфрақұрылымды жобалауда қауіпсіздік — күннен-күнге маңыздылығы артып келе жатқан өте күрделі мәселе. Дәстүрлі «сенімді шекара» қауіпсіздік үлгісі бұлттық ортада жеткіліксіз болып шықты, өйткені корпоративтік желілер мен деректер орталықтарының физикалық шекаралары бұлтта жойылып кетеді. Нөлдік сенім сәулеті осы мәселенің шешімі ретінде пайда болып, «ешкімге сенбе, барлығын тексер» қағидатына негізделді. Бұл тәсіл желінің «ішкі» немесе «сыртқы» болуын ескермей, барлық пайдаланушылар мен жүйелерді бірдей деңгейде тексеруді талап етеді.

Нөлдік сенім сәулеті АҚШ Ұлттық стандарттар және технологиялар институтының SP 800-207 стандартында егжей-тегжейлі сипатталған. Стандартқа сәйкес, нөлдік сенім үлгісінің негізгі қағидаттары: барлық ресурстарды желі ішкі немесе сыртқы екендігіне қарамастан қорғалмаған деп санау; тұтынушы жеке басын растауды және рұқсатын тексеруді барлық сұрауда орындау; деректерге кіруді «ең аз артықшылық» қағидаты бойынша шектеу; барлық желілік трафикті шифрлау; тұрақты қадағалау мен журналдауды жүзеге асыру [11, 7 б.]. Google компаниясының BeyondCorp бастамасы — нөлдік сенімнің тәжірибелік іске асырылуының ең белгілі мысалы, ол 2009 жылғы амалдық жүйе деңгейіндегі шабуылдан кейін VPN-сіз қауіпсіздік үлгісіне жаппай ауысудың нәтижесі.

Жеке бас растау және кіруді басқару — бұлттық қауіпсіздіктің бірінші желісі. Кіруді

басқару саясаттары кімнің, қандай ресурсқа, қандай шарт бойынша кіре алатынын нақты анықтайды. Кіруді басқаруды дұрыс баптаудың маңыздылығы зор, өйткені шамадан тыс кең берілген рұқсаттар бұлттық жүйелердегі оқиғалардың басты себептерінің бірі болып табылады. Кіруді басқаруды жобалауда принципіалды бірнеше ережені ұстану маңызды: өндірістік орта және даму ортасы үшін бөлек есепшоттар; қызметтік есепшоттарға тек қажетті рұқсаттарды беру; көп факторлы аутентификацияны барлық деңгейде міндетті ету. Рөлге негізделген қол жетімділікті басқару — нақты пайдаланушыларға рұқсат беруден гөрі, рөлдерге рұқсат беру — ірі ұйымдардағы кіруді басқарудың тиімді тетігі.

Деректерді шифрлау екі күйде жүзеге асырылады: тасымалдау кезінде және тыныштық күйде. Тасымалдау деңгейіндегі қауіпсіздік хаттамасының 1.2 немесе 1.3 нұсқасы желілік трафикті шифрлайды. AES-256 алгоритмі деректер қорларын, нысандық сақтауды және файлдық жүйелерді шифрлауда кеңінен қолданылады. Шифрлау кілттерін басқару қызметі кілттердің толық өмірлік циклін — жасалуы, ауыстырылуы, таратылуы, жойылуы — орталықтандырылған түрде бақылайды. Аппараттық қауіпсіздік модулі ең жоғары қауіпсіздік талаптары бар ортаға — қаржы секторы, мемлекеттік жүйелер — арналған физикалық кілт сақтау шешімін ұсынады. Шифрлау қауіпсіздіктің техникалық тетігі ғана емес, сонымен қатар «Жалпы деректерді қорғау ережесі» және Қазақстан Республикасының «Дербес деректер және оларды қорғау туралы» заңы сияқты заңнамалық талаптарды орындаудың міндетті элементі [12, 318 б.].

Бұлттық жүйелерде автоматты осалдықтарды сканерлеу үздіксіз кіріктіру және үздіксіз жеткізу желісіне кіріктірілген болуы тиіс. Статикалық қосымша қауіпсіздігін тексеру бастапқы кодтағы қауіпсіздік кемшіліктерін табады. Динамикалық қосымша қауіпсіздігін тексеру жұмыс жасап тұрған қосымшаны сырттан тексереді. Контейнер суреттерін сканерлеу — Trivy, Snyk, Clair — контейнер суреттеріндегі осал кітапханаларды анықтайды. «Солға жылжыту қауіпсіздігі» тәсілі — қауіпсіздік тестілеуді бағдарламалық жасақтаманы жасаудың ең ерте сатысынан бастауды ұсынады, өйткені өндірістік ортада табылған осалдықты жөндеудің шығыны даму кезеңімен салыстырғанда 30 есеге дейін артуы мүмкін. Бұл тәсілді енгізу ұйымдардың жалпы қауіпсіздік деңгейін айтарлықтай арттырады.

3.2 Бақылаулылық, қызмет деңгейі туралы келісім менеджменті және бұлттық шығынды оңтайландыру

Бұлттық инфрақұрылымның тиімді жұмысын қамтамасыз ету үшін жан-жақты қадағалау жүйесін жобалау қажет. Бақылаулылық — жүйенің ішкі күйін оның сыртқы шығыстары арқылы білу мүмкіндігі. Заманауи бақылаулылық үш тіректен тұрады: өлшемдер, журналдар және іздер. Өлшемдер жүйе сипаттамаларын уақыт қатарлары ретінде сақтайды. Prometheus — ашық бастапқы кодты уақыт қатарлары деректер қоры ретінде бұлттық ортада ең кеңінен таралған өлшем жинау жүйесі. Grafana Prometheus

және басқа деректер көздерімен жұмыс жасайтын жеткізу алаңы. Олар жиі Prometheus+Grafana жұбы ретінде бірге қолданылады және инфрақұрылым топтарына жүйенің нақты уақыт жағдайын толық суреттеуге мүмкіндік береді [13, 278 б.].

Журналдар жүйе оқиғаларының хронологиялық жазбасы болып табылады және инциденттерді тергеу, аудит, сәйкестікті тексеру үшін маңызды. ELK дестесі — Elasticsearch, Logstash, Kibana — немесе EFK дестесі — Elasticsearch, Fluentd, Kibana — микроқызмет ортасынан жиналатын үлкен көлемдегі журнал деректерін орталықтандырылған жерде жинау, индекстеу және іздеу мүмкіндігін ұсынады. OpenTelemetry — бұлтқа туған ортада бақылаулылық деректерін — өлшем, журнал, із — жинау үшін стандарттанған, қызмет берушіге тәуелсіз шеңбер. Бөлінген іздеу — бөлінген жүйелерде бір сұраудың бірнеше микроқызмет арқылы өту жолын ізтабу, кешігу мәселелерін анықтау үшін маңызды. Jaeger, Zipkin, AWS X-Ray — осы мүмкіндіктерді ұсынатын кең таралған алаңдар.

Ескерту жүйесі белгілі бір өлшемдер белгіленген шегінен асқанда немесе белгілі бір оқиға орын алғанда операторларды немесе автоматты түзету жүйелерін хабардар етеді. Ескерту баптауын жобалауда «ескерту шаршауы» мәселесін ескеру маңызды: тым көп жалған оң нәтижелер немесе маңызы аз ескертулер топтардың маңызды оқиғаларды жіберіп алуына алып келуі мүмкін. Сайт сенімділігі инженериясы тәжірибесіне сәйкес, ескертулер тек пайдаланушыға тиетін жағдайлар үшін берілуі тиіс. Ескертулер топшасы орта реакирлеу уақыты бойынша жіктелуі тиіс — шұғыл, ескерту, ақпараттық — және әрқайсысы үшін нақты жауапкершілік анықталуы тиіс.

Қызмет деңгейі туралы келісім — ұйым мен тұтынушы немесе ішкі ақпараттық-технологиялар топтары арасындағы өнімділік пен қол жетімділік міндеттемелерін анықтайды. Қызмет деңгейі көрсеткіші — нақты өлшенетін қызмет сипаттамасы, мысалы, 99-шы пайыздылықтағы кешігу немесе қол жетімділік пайызы. Қызмет деңгейі мақсаты — қызмет деңгейі көрсеткішіне қатысты мақсатты мән, мысалы, «99,9 пайыз уақытта жүйе қол жетімді болуы тиіс». Қате бюджеті — қалған қызмет деңгейі мақсаты жиынтығы болып табылады; топ қате бюджетін жаңа мүмкіндіктер шығаруға жұмсай алады, алайда бюджет тауысылған жағдайда сенімділік жұмыстарына назар аударуы тиіс. Бұл тәсіл Google-дың сайт сенімділігі инженериясы кітабында егжей-тегжейлі сипатталған және мыңдаған компания бүгінде осы методологияны қолданады [14, 56 б.].

Бұлттық шығынды оңтайландыру бұлттық инфрақұрылымды басқарудың стратегиялық бөлігіне айналды. Зерттеулер бойынша, орта есеппен ұйымдар бұлттық ресурстарының 30-35 пайызын пайдасыз немесе тиімсіз пайдаланады. Шығын оңтайландырудың негізгі тәсілдері: алдын ала брондалған инстанстар немесе жинақтау жоспарлары арқылы алдын ала міндеттеме алу — 1-3 жыл — талап бойынша бағамен салыстырғанда 40-70 пайыз жеңілдік береді; тапшы инстанстар — үзіліске төзімді жұмыс

жүктемелері үшін 70-90 пайыз жеңілдік ұсынатын артық бұлт ресурстарын пайдалану; дұрыс мөлшерлеу — нақты пайдаланылу деректеріне негізделіп инстанс түрі мен мөлшерін оңтайландыру; деректерді мұрағаттау — сирек қол жеткізілетін деректерді арзанырақ жинақтауыш деңгейлеріне жылжыту.

Қаржылық операциялар тәжірибесі — бұлт шығынын техникалық, қаржылық және кәсіпкерлік топтардың бірлескен жауапкершілігі ретінде қарастырады. Қаржылық операциялар шеңбері үш кезеңнен тұрады: хабардар болу, оңтайландыру және жұмыс жасау. AWS Cost Explorer, Azure Cost Management, Google Cloud Cost Management — негізгі бұлт қызмет берушілерінің жергілікті шығын талдау құралдары. Infracost, CloudHealth, Arptio Cloudability — көп бұлттық шығын менеджменті алаңдары. Бұлттық инфрақұрылымды жобалаудың ерте кезеңінен бастап шығын тиімділігін ескеру — кейіннен ірі шығындарды болдырмаудың негізгі жолы [15, 302 б.].

Кәсіпкерлік тұрақтылығын қамтамасыз ету үшін апатты қалпына келтіру стратегиясы жасалуы тиіс. Апатты қалпына келтіру жоспарлауда екі негізгі өлшем анықталады: қалпына келтіру уақытының мақсаты — апаттан кейін жүйені қалпына келтіруге рұқсат берілген ең ұзақ уақыт; деректерді қалпына келтіру нүктесінің мақсаты — деректер жоғалтудың рұқсат берілген ең үлкен мөлшері. Осы екі өлшемнің төмен болуы жоғары шығынды білдіреді, сондықтан кәсіпкерлік талаптармен теңестіру маңызды. Апатты қалпына келтіру стратегиялары шығын-тиімділік бойынша: сақтық көшірме мен қалпына келтіру, пилоттық жарық, жылы резерв, белсенді-белсенді көп тораптық деп жіктеледі.

Автоматты сақтық көшірмелеу жобаның ажырамас бөлігі болуы тиіс. «3-2-1 ережесі» — деректердің үш көшірмесі, екі түрлі тасымалдаушыда, олардың бірі географиялық алшақ орналасқан — бұлттық ортада да өзектілігін жоғалтпайды. Сақтық көшірмелердің тиімділігін тексеру үшін кезеңдік қалпына келтіру жаттығулары жүргізілуі тиіс. Netflix компаниясының «Бас тарту инженериясы» тәжірибесі — Chaos Monkey құралы арқылы өндірістік ортада кездейсоқ қызметтерді өшіріп, жүйенің төзімділігін тексеру — бүгінде бұлттық инфрақұрылымның сенімділігін тексерудің заманауи жолы ретінде кеңінен таралды. Бұл тәсіл болашақта нақты апат болғанда жүйенің дұрыс жұмыс жасайтынына сенімдерін арттырады.

ҚОРЫТЫНДЫ

Жүргізілген зерттеу бұлттық есептеулер негізіндегі ақпараттық инфрақұрылымды жобалаудың күрделі, көп қырлы үдеріс екенін айқын көрсетті. Табысты жобалау техникалық білімді ғана емес, сонымен қатар кәсіпкерлік талаптарды, шығын-тиімділік балансын және ұзақ мерзімді стратегиялық мақсаттарды жете түсінуді талап етеді.

Бірінші тарауда қарастырылған бұлттық есептеулердің тұжырымдамалық негіздері мен қызмет үлгілері — инфрақұрылым-қызмет ретінде, платформа-қызмет ретінде,

бағдарлама-қызмет ретінде, функция-қызмет ретінде — ұйымдардың алдына нақты таңдау мүмкіндіктерін қояды. Қызмет үлгісін таңдауда ұйымның техникалық жетілу деңгейі, бақылау мен икемділік арасындағы ымыра, сонымен қатар заңнамалық сәйкестік талаптары шешуші рөл атқарады. Орналастыру үлгілерінің — жалпы, жеке, будандастырылған, көп бұлт — алуандығы ұйымдарға өздерінің бірегей жағдайына сай оңтайлы шешімді таңдауға мүмкіндік береді.

Екінші тарауда талданған техникалық астарлар — микроқызмет сәулеті, Kubernetes контейнер үйлестіруі, бағдарламалық жолмен анықталатын желі сәулеті, инфрақұрылым-код ретінде тәсілі — заманауи бұлттық инфрақұрылымның базалық тіректері болып табылатыны анықталды. Инфрақұрылым-код ретінде қағидаты топ өнімділігін арттырып, қателіктерді азайтып, тексеру мүмкіндіктерін кеңейтеді. Автоматты масштабтау мен оңтайлы кәштеу тетіктері жоғары жүктемеде де жүйенің сенімді жұмысын қамтамасыз етеді.

Үшінші тарауда зерделенген қауіпсіздік сәулеті мен басқарылу мәселелері бұлттық жобалаудың ажырамас бөлігі екені белгілі болды. Нөлдік сенім үлгісі бұлттық ортаның сенімді емес желі сипатына бейімделген жаңа қауіпсіздік ойтолғауы болып табылады. Сайт сенімділігі инженериясы тәжірибесіндегі қызмет деңгейі көрсеткіші, қызмет деңгейі мақсаты, қате бюджеті жүйесі кәсіпкерлік талаптары мен инженерлік шешімдерді байланыстырудың тиімді тетігі. Қаржылық операциялар тәсілі бұлттық шығындарды кәсіпкерлік мақсаттармен теңестірудің жолы ретінде барған сайын маңызды болуда.

Жалпы қорытынды ретінде бұлттық инфрақұрылымды жобалаудың табысты болуы үш негізгі шартқа байланысты деп тұжырымдауға болады: кәсіпкерлік талаптарды жете түсіну, дұрыс сәулеттік шешімдер қабылдау және операциялық тиімділікті үздіксіз жақсартып отыру. Технологиялар тез өзгеретін бұл салада «мәңгі бір нұсқалы» шешімдер жоқ — инфрақұрылымды жобалаушылар жаңа технологиялар мен тәжірибелерді үздіксіз игеріп, жүйелерін жаңартып отыруы тиіс. Қазақстан Республикасындағы цифрлы ауысу кезеңінде отандық ұйымдардың бұлттық инфрақұрылымды дұрыс жобалау мәдениетін қалыптастыруы — ел экономикасының бәсекеге қабілеттілігін арттырудың стратегиялық бағыты болып табылады.

Пайдаланылған әдебиеттер тізімі

1. Байжанов А.Б., Сейткали Е.Ж. *Бұлттық технологиялар негіздері: оқулық.* — Алматы: «Дәуір» баспасы, 2020. — 248 б.
2. Mell P., Grance T. *Бұлттық есептеулердің NIST анықтамасы. NIST арнайы жарияланым 800-145.* — Вашингтон: АҚШ Ұлттық стандарттар және технологиялар институты, 2011. — 7 б.
3. Олифер В.Г., Олифер Н.А. *Компьютерлік желілер: принциптер, технологиялар,*

- хаттамалар: оқулық. — Санкт-Петербург: Питер, 2020. — 1008 б.
4. Қазақстан Республикасының «Ақпараттандыру туралы» Заңы. 2015 жылғы 24 қарашадағы № 418-V ҚРЗ. — Астана: ҚР Парламенті, 2015. — 68 б.
 5. Таненбаум Э., Уэзеролл Д. Компьютерлік желілер. 5-ші басылым. — Санкт-Петербург: Питер, 2019. — 960 б.
 6. Захаров О.И. Облачные вычисления в корпоративных информационных системах. — Москва: МГТУ им. Н.Э. Баумана, 2021. — 312 с.
 7. Newman S. Создание микросервисов. 2-е изд. / Пер. с англ. — Санкт-Петербург: Питер, 2022. — 624 с.
 8. Burns B., Beda J., Hightower K. Kubernetes: вверх и в путь. 3-е изд. / Пер. с англ. — Санкт-Петербург: Питер, 2023. — 368 с.
 9. Нигматуллин Р.Р., Жусупов А.Д. Деректерді сақтау жүйелері және мәліметтер қаупсіздігі. — Алматы: ҚазҰТЗУ баспасы, 2021. — 196 б.
 10. Ким Дж., Дебуа П., Уиллис Дж. DevOps-справочник. / Пер. с англ. — Москва: IT Revolution Press, 2019. — 536 с.
 11. Rose S., Borchert O., Mitchell S., Connelly S. Архитектура нулевого доверия. NIST SP 800-207. — Гейтерсберг: Национальный институт стандартов и технологий США, 2020. — 50 с.
 12. Исмаилов Т.А., Мамыров Е.Б. Ақпараттық қауіпсіздік және деректерді криптографиялық қорғау. — Алматы: Абай атындағы ҚазҰПУ баспасы, 2022. — 384 б.
 13. Сридхаран К. Наблюдаемость распределённых систем. / Пер. с англ. — Санкт-Петербург: Питер, 2021. — 216 с.
 14. Бейер Б., Джонс К., Петофф Дж., Мёрфи Н.Р. Site Reliability Engineering: как Google управляет производственными системами. / Пер. с англ. — Москва: Альпина Диджитал, 2019. — 672 с.
 15. Рахимов С.К., Аширбаев Б.Д. Бұлттық инфрақұрылымды ұйымдастыру және шығын менеджменті. — Нұр-Сұлтан: «Л.Н. Гумилев атындағы ЕҰУ» баспасы, 2023. — 228 б.

ҚМ АА Куәлік нөмірі: **KZ45VPY00102718** — ҚР Мәдениет және Ақпарат министрлігі

© 2026 **Bilimger.kz** Ақпараттық-танымдық білім порталы. Барлық мазмұн авторлық құқықпен қорғалған.