Добрый день, уважаемые студенты! Я рад вас всех видеть, приветствую вас на следующей нашей лекции. Мы переходим к первому модулю и сегодня мы с вами поговорим про основные понятия в облаке.

Итак, поехали. Наша сегодняшняя лекция поделена на четыре части, первые две части не привязаны к AWS, а больше про облачные вычисления, которые могут быть применены к любому облачному провайдеру. И так, первая часть у нас будет введение в облачные вычисления. Вторая часть про преимущества облачных вычислений. Далее, в третьей части у нас будет введение в AWS, т.е. Amazon Web Services и в четвёртой части мы поговорим про документ, который называется AWS Cloud Adoption Framework (AWS CAF).

Так начинаем. Первая часть. Введение в облачные вычисления. Что такое облачные вычисления? Облачные вычисления – это доставка по требованию IT ресурсов, такие как вычислительные мощности, базы данных, приложения, хранения и другие, через интернет по модели pay-as-you-go pricing. Pay-as-you-go pricing означает, что вы оплачиваете только за то и за то время, в которое вы использовали те или иные ресурсы. Эти ресурсы запускаются на серверных мощностях, которые находятся в больших дата-центрах, а эти дата-центры находятся в различных локациях по всему земному шару. Этими дата-центрами владеет облачный провайдер, в нашем случае AWS. Когда мы говорим про определённый сервис AWS, то следует понимать, что каждый сервис, в том или ином виде, использует серверные мощности, расположенные на одном из дата-центров. Также, следует помнить, что при AWS есть более 200 различных сервисов и вы должны воспринимать это, таким образом, как лего элементы, лего блоки, т.е. что-то вы используете для того, чтобы строить стены дома, например, что-то вы используете как дверь, что-то используете как окно, где-то это диван, и так далее. Таким образом, используя и компануя различные сервисы, вы можете построить одну целую IT-инфраструктуру. Облачные вычисления позволяют нам думать об инфраструктуре не как о железе, а как о каком-то софте. Что же это значит? Чтобы это понять, давайте сравним две различные вычислительные модели, первое – это традиционная, вторая – это облачная. Когда мы говорим про традиционную модель вычисления, то подразумевается локальный дата-центр и инфраструктура воспринимается как железо. Что это значит? Это значит, что вы владеете этим локальным дата-центром, вам нужно, чтобы у вас было физическое помещение, вам нужно, чтобы у вас был персонал, который обслуживает это помещение, оборудование, которое там стоит, это серверные мощности. Также вам нужно обеспечить физическую безопасность, так как это оборудование достаточно дорогостоящее. Планировать как минимум расположение, планировать расширение, либо наоборот уменьшение этого помещения. Это все подразумевает capital expenditures, т.е. капитальные вложения – это большие вложения, которые вам нужно внести, чтобы начать. Следующее, это долгий закупочный цикл железа, что это значит? Это значит то, что, если вам нужно докупить какое-то оборудование, то вам нужно его заказать, когда вы заказываете у поставщиков, в лучшем случае, она к вам приедет за несколько дней. В худшем случае, это может занять месяц. Далее, когда к вам приезжает это железо, вы её принимаете, настраиваете и подключаете к дата-центру. Это все происходит не в течение часа и даже не в течение полудня. Поэтому это достаточно долго, более того, вам нужно постоянно угадывать теоретически максимальные нагрузки. Что это значит? Это значит, что вам нужно просчитать те вычислительные мощности, которые достаточны для вашего бизнеса. Теперь, здесь могут возникнуть два момента, если вы владеете недостаточными серверными мощностями, то ваше железо не будет справляться с нагрузками. Менее приятный случай, это когда все те нагрузки, которые придут поверх, не обработаются, что уже не есть хорошо. В худшем случае, это когда вся инфраструктура упадёт и откажется работать, тогда вы все запросы на момент пика не сможете обработать. Обратный случай, когда вы закупились определёнными серверными мощностями и нагрузки небольшие, то все это время, когда у вас нагрузки небольшие, вы фактически переплачиваете, потому что вы купили серверные мощности слишком мощные. Это тоже проблема, это несёт большие затраты. Теперь подумайте о тех бизнесах, у которых нагрузка меняется в течение дня, для них ещё хуже. Давайте представим любой веб-сайт, магазин, который продаёт, допустим, бытовую технику и как мы все с вами знаем, ночью никто не закупается. В течение рабочего дня тоже никто не закупается, таким образом, нагрузка на сервисные мощности минимальная. Тогда когда у нас часы пика, т.е. это где-то начиная с 18 часов вечера локального до 21 часов вечера у нас максимальные нагрузки, люди начинают что-то покупать, таким образом, у вас нет такого, что у вас нагрузки постоянны. Это очень редко и в большинстве случаев, нагрузка она динамически меняется даже в течение дня и даже часа. Поэтому, даже в том случае, когда вы угадаете достаточно точно тот теоретический максимум, который у вас может быть, вы всегда имеете определённый период, даже в течение дня, когда у вас часть серверных мощностей просто не работает и зря простаивает. Это и есть те расходы, которые вы несёте в традиционной вычислительной модели.

Теперь, рассмотрим вторую модель – модель облачных вычислений. Она контрастирует с предыдущей моделью, и мы воспринимаем инфраструктуру как софт, как сервис. Это значит, что все может быть гибко изменено, и вы оплачиваете только за то, что вы используете, за то время, которое вы использовали определённый сервис или ресурс. Таким образом, вы можете настроить автоматическое изменение размеров инфраструктуры в зависимости от нагрузки на ваш сайт, приложение, ну и в целом на IT-инфраструктуру. Другой важный момент – это то, что вам нет необходимости гадать ту ёмкость, те серверные мощности, которые необходимы для вашего бизнеса. Этим никто не будет уже заниматься в облачной вычислительной модели, так как сервера могут не только людьми вручную в нужный момент, но и программно запускаться тогда, когда нужно для вашего решения, для вашего бизнеса, для вашей инфраструктуры. Это очень важно, потому что даёт достаточно большие преимущества: во-первых, вам не нужно содержать локальный дата-центр, вам не нужно содержать штат людей, который занимается не тем, чем занимается основной ваш бизнес, а это побочное направление, побочное управление, департаменты, которые поддерживают работу вашего бизнеса. Когда мы говорим про облачную модель, вам не нужно содержать этот штат, этих людей, вам не нужно искать помещение, вам не нужно отвечать за безопасность дата-центра, потому что дата-центр есть у AWS и мы в нужном количестве кусок этого дата-центра арендуем на нужное для нас количество времени.

По мере развития модели облачных вычислений появляются различные стратегии и модели предоставления сервиса, которые отличаются уровнем контроля, гибкости и управляемости. Давайте подробнее об этом поговорим. На текущий момент, выделяют 3 различные модели предоставления облачного сервиса. Они отличаются уровнем контроля над IT ресурсами. Первая – это IaaS, т.е. infrastructure as a service. Вторая – это PaaS, т.е. platform as a service. Самая последняя это – SaaS, т.е. software as a service.

Давайте поговорим подробнее про каждую из этих моделей. Начнём с IaaS, IaaS – это infrastructure as a service и подразумевается, что в этой модели у вас максимальный контроль над ресурсами, которые вам даются, а именно, это ресурсы на уровне сетей, т.е. вы можете настроить сети по вашему желанию на уровне самих серверов, т.е. вы настраивая сеть, говорите, что вот эти сервера у нас в этой сети, эти сервера в этой сети. Так они между собой взаимосвязаны и даже на уровне жёстких дисков, где хранятся ваши данные. Таким образом, IaaS максимально приближена к традиционной вычислительной модели, где у вас полный контроль: вы на сервера устанавливаете операционную систему, вы устанавливаете по необходимости приложения, которые вам нужны, вы сами поддерживаете и операционную систему и приложения, при появлении новых версии, обновлений, может быть даже критических обновлений безопасности, это все вам нужно делать самостоятельно. Максимально приближенная версия к традиционной вычислительной модели в облаке, это – модель предоставления сервиса по модели infrastructure as a service . Единственное отличие IaaS от традиционной вычислительной модели это то, что вы работаете и воспринимаете инфраструктуру не как железо, hardware, а как software, т.е. софт. Этим она и отличается.

Следующая модель – это PaaS, platform as a service. Один из примеров, это сервис AWS RDS, т.е. Relational Database Service, когда вам предоставляется база данных как сервис. Что это значит? Это значит, что вам не нужно устанавливать операционную систему, вам не нужно устанавливать приложения баз данных, т.е. СУБД (Система управления базами данных) на этот сервер, заниматься обновлениями и так далее, вам предоставляется база данных как сервис, все остальное операционная система, приложение, а также железо – на стороне облачного провайдера, т.е. вы видите, контроля у вас уже меньше и какую-то часть облако берет на себя. Если у вас вот работает допустим база данных, если у вас ломается железо, что-то с ним там происходит, то прозрачно для вас, облачный провайдер его заменит и вы продолжите работать с этим сервисом. Если же выходит, например, какое-то критическое обновление для операционной системы либо для СУБД, то это тоже происходит прозрачно для вас, облачный провайдер за вас это делает, таким образом, вам содержать администратора баз данных уже нет необходимости, вам нужны разработчики и разработчики баз данных, которые работают с данными внутри этой базы данных, строят модель данных и так далее. Вы видите, контроля меньше, но, тем не менее, когда мы говорим platform as a service, это те сервисы, которые не являются конечным продуктом, а в большинстве случаев, являются частью одной целой IT-инфраструктуры.

Последняя модель предоставления облачного сервиса – это SaaS, software as a service. Отличительной особенностью этой модели является то, что сервис предоставляется как готовый продукт, у вас минимальное количество контроля, вы, скорее всего, не сможете его как-то модифицировать и можете использовать только те настройки, которые облачный провайдер для вас вывел наружу. Самым ярким примером software as a service является электронная почта. Например, Gmail или WorkMail, есть у Amazon такой cервис почты. Фактически, это готовый продукт, который могут использовать конечные потребители. Тем не менее, мы говорили, что каждый сервис может являться отдельным блоком в вашей IT-инфраструктуре и даже сервисы SaaS могут быть частью этой инфраструктуры и использоваться как одно целое. Я надеюсь, вам понятна разница между этими всеми моделями. Самый основной момент это то, что начиная от IaaS опускаясь ниже через PaaS до SaaS, вы видите, что контроля над IT ресурсом у вас меньше. То, что контроля меньше, не говорит о том, что это плохо. Это говорит о том, что вам не нужно заниматься какой-то частью и в зависимости от ваших бизнес потребностей, вы выбираете тот или иной сервис, например, если вы какая-то компания, которая занимается сложными вычислительными расчётами, например, научными, то вам, скорее всего, нужна не простая база данных, а какая-то продвинутая с определёнными настройками, специально оптимизированными для вашего типа данных, которые там крутятся. В этом случае, обычный PaaS, т.е. сервис AWS RDS вам не подойдёт, потому что там есть какие-то определенные настройки, вы что-то можете поменять, но полного контроля у вас не будет, поэтому ваш вариант – это IaaS. В случае, когда вам достаточно стандартных настроек или нужно в небольшом количестве, не сильно оптимизировать ваш сервис, вашу базу данных, то вам достаточно использовать готовый сервис AWS RDS как пример, и не содержать целого системного администратора, который занимался бы узконаправленной работой и обслуживал бы эту базу данных, за вас это делает облачный провайдер. Он делает это автоматически. Там сильные специалисты, которые собрались со всего мира и сделали предварительно эту настройку в лучшем виде для вас – это первое, второе – исключается человеческий фактор, т.е. вряд ли что-то там поломается из-за того, что не такую настройку сделали. Она автоматически для вас развёртывается, т.е. поднимается и даётся как сервис. Когда это у вас локальный, живой физически, системный администратор, всегда есть человеческий фактор, где-то что-то поломалось и оно может больно отразиться на вашем бизнесе. Поэтому, в случае, когда вам нужно меньше контроля для базы данных, как пример, вы можете воспользовать PaaS. Последний случай, это когда вам нужно что-то максимально готовое. У вас, например, нет соответствующих специалистов для поддержания вот определённого софта, но при этом этот софт является частью вашей инфраструктуры, то вы можете посмотреть в сторону SaaS сервисов, которые представлены как готовый продукт. Самый популярный пример – это веб-почта. Другие примеры, касающиеся каждого из видов моделей предоставления облачного сервиса, мы будем рассматривать на следующих занятиях.

Теперь, давайте подробнее поговорим о том, какие модели развёртывания IT-инфраструктуры существует на сегодняшний день. Это – облачная, гибридная и on-premises, т.е. локальная, или по-другому ещё называют private cloud. Давайте разберём, чем каждая отличается. Когда мы говорим про облачную, cloud deployment model, это значит вся ваша IT-инфраструктура хостится на одном или нескольких из облачных провайдеров. Далее, когда мы говорим on-premises, это значит локальный дата-центр или локальный провайдер дата-центра, услугами которых вы пользуетесь, но суть в том, что вся ваша инфраструктура развёрнута в черте вашего города, либо региона и не является хостер, будь то вы, либо это локальный провайдер не является публичным облачным провайдером. Есть гибридный вариант, когда вы хостите часть инфраструктуры в облаке и часть хостите локально у себя. Это достаточно не распространённая, но и не такое редкое явление, потому как те компании, которые изначально были локальные начинают переходить в облако. В процессе перехода они работают с гибридной инфраструктурой до того момента, пока они полностью не перейдут в облако, т.е. это может быть промежуточным этапом, а может быть и конечным состоянием, когда у вас в облако, например, отправляются бэкапы ваших баз данных и вся инфраструктура хостится на локальном дата-центре по тем или иным причинам. У вас может сложиться впечатление, что облачная IT-инфраструктура и локальная IT инфраструктура абсолютно разные вещи, на самом деле под капотом одно и то же, а именно сервера обычно установлены Linux, поверх него устанавливаются различные приложения, которые составляют элементы, блоки одной целой нашей IT-инфраструктуры. Единственное отличие это то, что в облаке вы можете добраться до ваших IT ресурсов на более высоком абстрактном уровне и они обёрнуты, представлены вам в виде AWS сервисов. Когда мы говорим про локальную IT-инфраструктуру, то вы с этими IT ресурсами взаимодействуете напрямую, т.е. с железом, сами его подключаете к локальной сети, дальше устанавливаете операционную систему, приложения и все все все делаете самостоятельно. Я бы хотел сейчас на этом слайде пройтись по моментам, что есть в локальной инфраструктуре и как это называется на стороне облака. Мы рассмотрим различные группы IT ресурсов, это безопасность, сети, вычислительные ресурсы и хранение и базы данных. Когда мы говорим про безопасность, мы подразумеваем на локальной IT-инфраструктуре Firewalls, ACLs, Administrators. А когда мы говорим про безопасность в облаке, то примерами из сервисов на стороне облака, связанной с безопасностью являются, например, Security groups, Network ACLs, IAM сервис и так далее. Когда мы говорим про сети, например, локально мы говорим у нас есть роутер, у нас есть Network pipeline, у нас есть Switch, то в облаке есть аналоги, сервисы, которые позволяют делать то же самое, как пример Elastic Load Balancing, Amazon VPC. Следующая группа – это вычислительные мощности. На локальной IT-инфраструктуре это вот ваше железо отдельный сервер, на котором можно что-то установить, что-то с ним сделать. На стороне облака физически до этого сервера, конечно, вы добраться не сможете, но вы сможете его выделить под себя и уже работать с ней так же как вы работаете с локальным сервером. Следующее – это хранение и базы данных, т.е. на стороне локальной IT-инфраструктуры, вы настраиваете различные типы хранения данных, как DAS, SAN, NAS, также мы работаем напрямую с системами управления базами данных, т.е. RDBMS (relational database management system). На стороне облака у нас есть полные аналоги сервисов, такие как Amazon EBS (Amazon Elastic block storage) для блочного хранения, Amazon EFS (Amazon Elastic File System) это тоже блочное хранение, но есть возможность предоставить общий доступ нескольким серверам к одному блоку хранения, тоже блочное хранилище. Дальше, мы говорим про Amazon S3, если нам нужно объектное хранилище. Ну и последняя уже знакомая вам, это Amazon RDS, т.е. Amazon Relational Database Service, для того, чтобы работать с базами данных. Это я к тому говорю, что для любого IT ресурса в традиционной локальной IT-инфраструктуре можно подобрать один или несколько аналогов сервисов в облаке AWS. Мы подошли к заключительному слайду первой части нашей сессии, и я бы хотел пройтись по основным моментам, которые вам нужно запомнить и которые вам помогут, могут прийти в виде вопроса на реальном экзамене. Хорошо. Первое – это то, что такое облачные вычисления? Облачные вычисления – это доставка по требованию IT ресурсов через интернет по модели pay-as-you-go pricing, т.е. плати только за то, что ты используешь. Второе – это то, что облачные вычисления позволяют нам думать об инфраструктуре не как о железе, а как о software, т.е. как о сервисе, либо софте. Далее. Всего существует три модели предоставления облачного сервиса это: первая – это IaaS, т.е. infrastructure as a service. Вторая – это PaaS, т.е. platform as a service. Третья это – SaaS, т.е. software as a service. Далее, выделяют три варианта развёртывания IT-инфраструктуры это cloud, полностью инфраструктура хостится у облачного провайдера. Гибридная – это когда часть вашей IT инфраструктуры находится в облаке, а часть локально. И on-premises, т.е. вся ваша IT инфраструктура хостится на локальном дата-центре. Последнее, что следует помнить, что существует большое количество сервисов AWS и вы фактически на любой IT ресурс, который есть на локальном дата-центре, можете найти один или несколько вариантов сервисов, которые будут делать то же самое.

Мы переходим ко второй части нашей сессии и поговорим про преимущества облачных технологий. Вы можете остановить это видео и подумать, почему же большинство компаний во всем мире начинают переходить в облако. Большинство ответов у вас уже есть и на следующих слайдах мы рассмотрим шесть преимуществ, которые дают облачные технологии.

Первое преимущество облачных технологий – это вы обмениваете CapEx, т.е. Capital Expenses или капитальные расходы на Variable Expenses, т.е. расходы, которые у вас могут меняться. Что это значит? Это значит, что вам нет необходимости вливать большие вложения для того, чтобы начать ваш бизнес. Для того, чтобы построить IT-инфраструктуру, вам не нужно поднимать дата-центр, вам не нужно покупать это здание, вам не нужно покупать дорогостоящее оборудование. Это все вы обходите стороной и вы с использованием облачных технологии и соответствующих сервисов, арендуете на какое-то время ровно то, что вам нужно. В тот момент, когда это вам становится не нужно, вы её передаёте обратно облачному провайдеру и за это больше не оплачиваете. Таким образом, вам не нужно, когда вы начинаете создавать IT инфраструктуру, нести капитальные большие расходы, которые вы на тот момент используете. Это очень важно для бизнеса, потому что когда бизнес только-только начинается, либо начинается какое-то новое направление, капитальные расходы это то, что замораживает финансовые средства, т.е. вы большое количество денег потратили на капитальные вложения. Нужно ещё денег, либо если у вас есть какой-то определённый бюджет на ваш проект, то меньше денег останется на то, чтобы потратить на другие ваши расходы. Это может отразиться на скорости, на качестве того продукта, который вы делаете или проекта. Тогда, когда мы говорим про облачные технологии, вы освобождаетесь от этих капитальных вложений, у вас фактически не замораживаются деньги и тот бюджет, который у вас выделен, вы можете потратить больше на другие ваши статьи расходов, либо же добиться абсолютно того же, но с меньшим бюджетом, потому что вы не несёте этих больших расходов.

Второе преимущество облачных технологий – это преимущества от массовости, масштабов этих развёртываний дата-центров, т.е. AWS экономит на том, что он открывает большое количество дата-центров, очень много железа задействовано в этом и за счёт того, что это происходит массово в большом объёме, где-то они получают скидки, где-то AWS выгоднее, потому что, представьте, вот у вас есть один сервер и вы нанимаете целого системного администратора, чтоб его администрировать. Тогда вся ваша зарплата фактически уходит в один сервер, чтобы поддерживать этот сервер и представьте системного администратора, который поддерживает сто серверов, это значит, его зарплата равномерно распределяется по вот этим ста ресурсам, таким образом, доля зарплаты этого человека размывается по вашей IT-инфраструктуре. Здесь похожий аналог: за счёт того, что они могут размазать часть своих расходов, облачный провайдер, он экономит на этом, а это все отражается на том, что он может предоставить для нас цены за сервисы намного ниже, чем если бы мы это делали самостоятельно своими силами. Таким образом, мы платим меньше и платим ещё меньше, потому что мы работаем по модели pay-as-you-go pricing, т.е. оплачиваем только то, что мы используем.

Следующее преимущество – это то, о чем мы с вами уже говорили. С облачными технологиями нет необходимости угадывать какой capacity, какая ёмкость или какие серверные мощности нам нужны для того, чтобы удовлетворять потребности бизнеса. Это на самом деле, достаточно серьёзная задача, с которой по определённым причинам не всегда удаётся правильно совладать. Что я имею в виду? IT департаменты во всех компаниях сталкиваются с такой проблемой, что им нужно рассчитать серверные мощности на текущий момент сколько нужно и планировать на будущее с учётом того, что поставка оборудования занимает какое-то время. Помимо поставки, его нужно принести, настроить, подготовить и подключить к IT-инфраструктуре, это все время, это все занимает несколько дней, недель, поэтому есть отдельный человек, либо штат, несколько людей, которая занимается этим планированием и делает прогноз, что через шесть месяцев мы растём такими темпами и нам нужно такое количество оборудования заказать к этому дню, например, либо через годьмы вырастем по текущим метрикам на такой объем, либо наоборот, уменьшимся и нам нужно столько-то железа докупить. Сам бизнес это такой род деятельности, который сложно прогнозировать, вы не знаете вы через полгода вырастите в три раза, в пять раз, а может в сто раз или, может быть, совсем закроетесь, через год это ещё сложнее планировать, как-то предугадать, что произойдёт. Поэтому всегда будет элемент такой, что не всегда удаётся точно определить, какое количество серверных мощностей вам нужно. Более того, у большинства бизнесов нагрузка на их IT-инфраструктуру неоднородна в течении дня, это тот пример веб-сайта бытовой техники, который мы ранее обсуждали. И это тот случай, когда нагрузка на IT-инфраструктуру меняется в зависимости от части дня и, например, в ночные часы, когда все спят, никто не заказывает, либо в рабочие часы, когда все работают, все ваши сервера не загружены или минимально загружены, фактически часть серверов простаивает. Другой случай, когда вы не рассчитали, либо у вас в какой-то момент либо проводится акция, либо по тем или иным причинам, пришёл большой трафик, большое количество пользователей и идёт увеличенное количество объем нагрузки на ваши сервера, то мы говорим это Underestimated server capacity, когда у вас сервера перегружены. В локальной IT-инфраструктуре вы с этим постоянно будете сталкиваться, и когда у вас нагрузки меняются в течение дня, то вы где-то упускаете возможную прибыль, потому что часть пользователей, которые хотели заказать что-то, не смогут заказать, потому что инфраструктура не может обработать этот запрос, она и так перегружена. Обратный случай, когда сервера просто простаивают, вы просто так, фактически, впустую оплачиваете за электричество, за аренду, именно за эти часы, и оборудование просто простаивает. Такого не происходит у нас в облаке и вы зависимости от требования к нагрузке к вашей инфраструктуре, можете настроить её таким образом, чтобы она автоматически менялась, она увеличивается в сто раз, когда это нужно и уменьшается в десять раз, если того требует нагрузка. Таким образом, вы несёте расходы в точности такие, которые вы используете. Это и есть третье преимущество облачных технологий – вы перестаёте гадать.

Следующие преимущества облачных технологий – это увеличенная скорость и гибкость. Что под этим имеется в виду? Мы тоже про это уже проговорили. Идея в том, что когда мы работаем с локальной IT-инфраструктурой как железо, мы сами делаем поставку оборудования, это может занимать дни и недели. Тогда как на стороне облака, если вам нужен тот или иной сервис, либо сервер, то они находятся у вас на расстоянии нескольких кликов, вы прокликаете и в течение нескольких минут получаете этот сервер. Таким образом, все то, что в обычное время без облачных технологий происходило бы недели, сейчас вы будете проделывать и получать за считанные минуты. Это просто сумасшедшее ускорение, которое даёт облачные технологии, позволяет вам, компаниям, бизнесу уменьшить затраты, которые несёт компания за счёт вот этого ускорения и других преимуществ облачных технологий. Дешевле, эффективнее и быстрее проводить различного рода эксперименты, например, когда запускается новый продукт или, в целом, когда вы делаете новое что-то, изначально основной ваш продукт, это все будет для вас намного дешевле и быстрее.

Пятое преимущество – это то, что мы перестаём напрямую взаимодействовать с железом, дата-центрами и не теряем на это ни времени, ни наши деньги. Поэтому бизнесу легче сконцентрироваться не на чем-то косвенном, а именно, на поддержке дата-центров, которые поддерживают основное решение, а сконцентрироваться на основном решении, на своих клиентах.

Самое последнее преимущество, которое так или иначе мы затрагивали – это то, что мы можем стать глобальными за минуты. Имеется в виду, наша глобальная инфраструктура AWS, которая находится в разных частях нашего земного шара, в различных локациях и даёт нам возможность строить IT-инфраструктуру, как результат вот этого распространения по всему земному шару становиться ближе к нашим клиентам и минимально нести за это затраты.

На этом мы добрались до последнего слайда второй части нашей сессии. Давайте пройдёмся по основным моментам, это шесть преимуществ, которые мы прошли. Первое – это то, что мы минимизируем капитальные расходы, либо совсем исключаем и переводим в расходы непостоянные, variable, которые пропорционально будут расти или уменьшаться в зависимости от того, как мы используем сервисы AWS. Второе – это массовость развёртывания этого решения, которое позволяет снизить цену сервиса для конечных пользователей, т.е. для нас, как разработчиков бизнеса, компаниям и так далее. Третье – это то, что мы перестаём угадывать ёмкость, т.е. нам не нужно угадывать какой объем вычислительных мощностей нам нужно. Это все делается автоматически на стороне облака. Четвёртое – это то, что мы получаем большую скорость и гибкость за счёт того, что мы не работаем с инфраструктурой как с железом, а работаем как с сервисом и все, что мы хотим запросить мы получаем за минуты, а не за неделю. Пятое – это то, что мы перестаём заниматься косвенными работами, а именно обслуживание и все работы, связанные с поддержкой дата-центров, а эти работы отдаём облачному провайдеру, а сами имеем возможность весь фокус, все внимание обратить на основной продукт и на наших клиентов. Последнее, шестое – это то, что AWS распространён по всему земному шару, есть дата-центры в различных локациях, что даёт нам возможность поднять инфраструктуру в любом месте и в итоге, быть ближе к нашим клиентам и пользователям.

Мы переходим к следующей части нашей сессии, это – введение в AWS, в этой главе мы рассмотрим его более подробно. В целом, что такое веб-сервис? Это – такая служба, доступная в интернете и даёт возможность вызвать себя через определённый стандартизированный формат общения, это, может быть, XML либо JSON. Таким образом, вы делаете запрос, получаете ответ и, другими словами, веб-сервис это некий API (application programming interface), с которым вы можете общаться, т.е. программы общаются между собой через API. Теперь, попробуем дать определение, что такое AWS, т.е. Amazon Web Services – это безопасная облачная платформа, которая объединяет все эти веб-сервисы в одном месте и даёт вам возможность запрашивать IT ресурсы по необходимости, по требованию. Вы платите только за то, что вы используете. Только за то время, в которое вы это все использовали и самое главное это то, что вы можете запросить любой ресурс, таким образом, вы можете в облаке с облачным провайдером построить IT-инфраструктуру любой сложности, правильно скомпоновав эти сервисы между собой. Как уже ранее упоминалось, у AWS есть более 200 различных сервисов. Эти сервисы поделены на различные категории, которые вы можете видеть на слайде. В каждой категории может быть от одного до десятка сервисов. В зависимости от ваших бизнес потребностей, вы ищете соответствующий вам сервис, изучаете её и используете в вашей IT-инфраструктуре. На этом слайде вы можете видеть простейший пример IT-инфраструктуры в облаке. Здесь представлены сервисы из четырёх различных групп сервисов, это: AWS VPC, т.е. Virtual Private Cloud – это сеть в облаке для того, чтобы все ваши ресурсы хранились в одном месте безопасно. Следующее это Amazon EC2 – это фактически ваш сервер в облаке. Он относится к группе сервисов Compute, т.е. вычисление. Представим, что там у вас крутится какое-то приложение, пользователь вызывает это приложение, и вы генерируете на каждого пользователя какой-то файл. Этот файл вам нужно хранить объектно, поэтому вы используете следующий сервис Amazon S3 – это сервис хранения для того, чтобы вот эти файлы в объектном виде сохранить в определённом месте. Также, вы можете подключить сюда сервис, например, Amazon DynamoDB – система управления базами данных класса NoSQL из группы сервисов Баз данных. В нём можете сгенерировать индекс для того, чтобы приложение быстрее находило ваши файлы, загруженные в сервис Amazon S3. Этот слайд был специально показан вам как пример того, что вы можете компоновать между собой эти сервисы, они могут между собой работать, и, скорее всего, ваша архитектура будет намного сложнее. Как я уже ранее вам говорил, что, в зависимости от ваших бизнес требований, либо требований к приложению, к IT-инфраструктуре, выбор сервисов в облаке будет отличаться от случая к случаю, поэтому ваша задача понимать, что за сервис, что он даёт, а дальше уже понимать для чего он используется. В тот момент, когда вам нужно для решения задачи использовать этот сервис, вы изучаете его глубже. Здесь вы можете видеть пример сервисов, которые есть, это далеко не полный список. Интересно то, что название сервиса, в большинстве случаев, начинается либо с Amazon, либо AWS, тут нет никакой абсолютной логики, мне интересно было как-то с этим разобраться, но никакого объяснения этому нет. Я объяснил для себя это тем, что AWS команда компании очень большая, в ней много команд и фактически для каждого сервиса выделена своя команда, которая разрабатывает и поддерживает. Я так понимаю, что название сервиса определяется именно этой командой, естественно она кем-то утверждается, но как предложение, изначально, она исходит из самой команды. Поэтому это такое хаотичное явление, здесь нет никакой логики и некоторые сервисы правильно пишутся как, например, Amazon EC2 или, например, Amazon Lightsail, а другие сервисы начинаются с AWS, такие AWS Lambda, AWS Fargate и так далее. Мы сейчас не будем останавливаться подробно на каждом из этих сервисов, потому что у нас будет возможность с ними познакомиться на следующих лекциях более подробно так, чтобы это лучше для вас запомнилось. Здесь основная идея была в том, что опять же, в зависимости от вашей необходимости для бизнеса, вы подбираете нужный для вас сервис. Для успешной сдачи экзамена AWS Certified Cloud Practitioner, вам нет необходимости знать все двести сервисов AWS. Для того чтобы успешно сдать экзамен, вам достаточно знать какую-то определённую часть из групп сервисов, указанных на слайде. Это ровно те группы сервисов и сервисы, которые мы будем рассматривать на нашем курсе. Я вам рекомендую это видео поставить на паузу и этот слайд внимательно один два раза прочитать. Да, вы многое не поймёте, но как минимум глаза начнут привыкать к названиям сервисов и меньше уже будет стресса от того, что слишком много чего нового и на следующей лекции, возможно, в каком-то смысле, будет легче для освоения.

Давайте остановимся теперь на том, как же взаимодействовать с этим огромным количеством сервисов у облачного провайдера. Всего существует три варианта: первый – это AWS Management Console. Второе – это AWS Command Line Interface (AWS CLI) или командная строка. Третье – это Software development kits (SDKs), проще говоря, это библиотека, которую вы подключаете, когда вы работаете с определённым высокоуровневым языком программирования и работаете с этой библиотекой, а эта библиотека делает соответствующие вызовы в облако для того, чтобы выполнить ту или иную операцию. Например, если мы хотим поднять один сервер, чтобы это сделать в AWS Management Console, будет отдельная тема, но я сейчас вкратце расскажу, чтобы вы поняли различия. Вы открываете, логинетесь в веб-сайте, это и есть ваша консоль. Там находите в списке, либо пишете в поисковой строке Amazon EC2. Открывается сервис Amazon EC2, там нажимаете на кнопку Create instance, открывается мастер создания инстанса и вы вводите необходимые параметры для его создания. Проходите через один или несколько шагов, т.е. страниц, заполняете параметры и в итоге этот сервис у вас появляется и там же в сервисе Amazon EC2, если вы откроете вкладку со списком инстансов, серверов, которые у вас есть, вы его там увидите, т.е. он будет создан, все есть. Как это сделать на стороне AWS Command Line Interface, командной строки, вы находите какая командная строка, какая команда вам нужна, в нашем случае это будет AWS, это общее название, пробел, EC2, пробел, Create instance, и задаёте здесь же, либо JSON, либо через аргументы в этой же строке, те параметры, которые нужны для создания инстанса. Нажимаете Enter, эта команда уходит и команда вызывает нужный API, и этот API создаёт вам инстанс, т.е. все то, что вы проделываете в консоли и то, что вы это написали как одну команду, что тот, что этот, они идут в одно место, они вызывают AWS API конкретного сервиса и конкретную операцию по созданию сервера. Теперь, когда мы говорим про Software development kits – это, как я ранее сказал, библиотека, например, если вы работали с языком программирования Python, то там есть библиотеки, вы подключаете библиотеки, какие-то библиотеки сторонние, какие-то уже есть предустановленные, и вы можете с ним работать для того, чтобы, например, работать с датами, с файлами, с операционной системой, с картинками. Теперь то же самое, есть специально разработанная библиотека, Boto3 для языка Python, чтобы вы могли взаимодействовать с облачным провайдером AWS. Как это происходит: вы подключаете библиотеку, например, в Python, это Import Boto3 и дальше вы создаёте сессию boto3.client название сервиса и сохраняете в переменной этот объект, этот объект это и есть ваш клиент, через этого клиента вам становятся доступны в зависимости от выбранного сервиса те или иные операции. Фактически, вы пишете код, какую-то сложную логику, и в этой сложной логике вы можете вызвать, например, то же самое создать/поднять инстанс для того, чтобы с ним потом что-то сделать и когда вы запускаете этот код, он у вас поднимает этот сервер. Теперь, ещё один важный момент то, что библиотеки, т.е. SDK существуют не только для одного языка программирования, а для нескольких, для многих популярных. Какие языки поддерживаются, вам как задание, да вы можете сейчас поставить на паузу это видео и попытаться сделать запрос в Гугл, либо в любом другом поисковике, такой, чтобы вам открылась страница, какие языки программирования поддерживаются в SDK. Окей, я надеюсь у вас получилось это сделать, у кого получилось молодцы, у кого не получилось, то мы ещё подобное будем проделывать на наших демо-сессиях. Хорошо, мы разобрали, я надеюсь, вам стало понятно, чем отличается консоль от командной строки и чем отличается SDK.

Вы как начинающие облачные инженеры, облачные специалисты в определённых пропорциях будете использовать каждый из них по мере продвижения по карьерной лестнице, например, сейчас представьте, что вы Junior специалисты, или даже до Junior, стажёры. Как пройдёте этот курс можно вас назвать Junior специалистами, после, получая больше опыта, вы станете Middle, Senior и дальше. В зависимости от того, как вы будете расти, время, проведённое с каждым из вариантов взаимодействия с AWS, будет меняться, давайте я поясню. Когда вы только-только начинаете, вам важно понять и представить у себя в голове, как этот сервис выглядит, какие у него возможности есть, и чтобы наглядно это посмотреть, правильней использовать AWS Management Console. Я так предполагаю, что 90% времени, когда вы будете работать с облаком, вначале вы будете работать в консоли, остальные по 5% это командная строка и SDK. По мере того, как вы будете расти, чуть позже, вы потихоньку будете наращивать долю времени, когда вы взаимодействуете с облаком, либо с командной строкой, либо с SDK для автоматизации вашей работы. Таким образом, вам не нужно уже самостоятельно ходить и вручную по одному создавать ресурсы в облаке, а вы будете использовать, либо набор командных строк, команд в командной строке, извиняюсь за тавтологию и запускать их разово, для того чтобы сделать какую-то операцию, не единичную, не атомарный, а какую-то более-менее сложную. Более продвинутый вариант, допустим, вы пишете какое-то приложение, которое использует облако. В этом случае, возможностей командной строки может быть недостаточно, командная строка идеально подходит для автоматизации вашей работы, если вы системный администратор. Если же вы больше разработчик, то, скорее всего, ваш потенциал умение программировать на более высокоуровневом языке программирования раскроет именно SDK. Как разработчик вы будете больше времени проводить с ним, писать более сложный код и таким образом, взаимодействовать с облаком. По ходу количество времени, когда вы используете консоль максимально минимизируется, допустим, я по себе скажу, вначале использовал консоль очень много, а сейчас я открываю консоль только тогда, когда мне нужно изучить какой-то новый сервис, я открываю, смотрю как выглядит, понимаю, что он может делать, представляю у себя в голове и дальше уже мне легче использовать этот сервис в командной строке или больше там, где я работаю, это SDK.

Хорошо, мы добрались до конца третьей секции нашей сегодняшней сессии. Давайте пройдёмся по основным моментам. Первая – это что такое AWS. AWS – это безопасная облачная платформа, которая объединяет большое количество сервисов, которые, в свою очередь, помогают вам создавать и работать с IT ресурсами всех возможных видов. Следующее, это то, что у нас сервисы объединены в различные группы и в зависимости от ваших бизнес потребностей, вы подбираете нужную группу, в нужной группе подходящий сервис и с ним работаете. Все сервисы AWS комбинируются между собой для того, чтобы помочь вам построить IT-инфраструктуру абсолютно любой сложности. Последнее – это то, что у нас есть три варианта взаимодействия с облаком: это AWS Management Console. Второе – командный интерфейс, командная строка, AWS Command Line Interface (AWS CLI). Третье – это Software development kits (SDKs), т.е. библиотека, которую вы подключаете внутри вашего языка программирования и можете с ним таким образом взаимодействовать внутри кода.

Мы добрались до последней части нашей сессии первого модуля и проговорим мы про AWS Cloud Adoption Framework (AWS CAF). Для сдачи экзамена подробно знать, что это такое вам не нужно. Это нужно будет вам в будущем, когда вы начнёте работать. Ещё больше вам нужно будет, когда вы станете архитекторами, либо техническими директорами в компаниях, у которых IT-инфраструктура в облаке или если компания сейчас находится локально и захочет в будущем переместиться в облако. Таким образом, это тот документ, который вам помогает максимально раскрыть те возможные моменты, которые надо учитывать при переходе с локальной IT-инфраструктуры в облачную. Когда мы говорим про переход с локальной IT-инфраструктуры в облако, мы должны рассмотреть весь этот процесс с нескольких сторон. Выделяют две группы: первая – это со стороны бизнеса, вторая – это со стороны технологии. Когда мы говорим про бизнес, она состоит из трёх частей, это собственно сам бизнес, ваши люди, специалисты, которые работают в вашей компании, и governance, т.е. управление, руководство. Когда мы говорим про технологии, мы рассматриваем платформу, безопасность и operations, т.е. операционную деятельность. На следующих слайдах мы поговорим более подробно о каждой из этой перспективе т.е. с определённой стороны и разберём её на более мелкие части, чтобы вам было понятнее, о чем мы говорим. В тот момент, когда команда, компания учтёт все моменты, которые могут возникнуть со стороны шести этих перспектив, можно говорить о том, что вероятность того, что компания успешно перейдёт в облако и максимально получит ту выгоду, те преимущества, которые облако даёт, будет увеличена.

Первая перспектива, т.е. взгляд, угол, с которого мы смотрим, это со стороны самого бизнеса. Здесь выделяют четыре части: это финансы, IT стратегия, потом реализация вот этих преимуществ, которые мы получаем и риски, с которыми работает бизнес. Что здесь важно? Важно то, чтобы сам процесс, сама вот эта инициатива перехода в облако, максимально поддерживалась и поощрялась высшим руководством, т.е. CEO, коммерческий директор, технический директор, потом бизнес менеджера, финансовые менеджера, любые эти должности, которые так или иначе влияют на бизнес, должны принять вот эту инициативу и соответствующе приоритезировать у себя, в своих планах эти моменты и учитывать эти изменения. Таким образом, направление самого бизнеса должно быть в одном направлении, куда смотрят IT, в нашем случае IT переходит в облако.

Когда мы смотрим на процесс перехода в облако со стороны People perspective, т.е. со стороны людей, специалистов, которые работают в этой компании, выделяют пять подпунктов: это Resource management, Incentive management – это стимул, управление стимулами, мотивации, Career management, соответственно, при построении карьеры весь этот процесс, тренинги и организационные изменения. Если говорить, опять же, вкратце, большая работа будет проделана со стороны HR-департамента. Он должен учитывать и понимать, что мы переходим в облако и сделать со своей стороны определённые действия, а именно, во-первых, убедиться, что у нас есть достаточные навыки у персонала, чтобы успешно быть в облаке. Если же нет, то подготовить и устроить процесс подготовки, тренировки, взращивания этих специалистов, которые умеют работать с облаком. Другой момент, специалисты бывают разные, например, не все из специалистов захотят, по тем или иным причинам, переходить в облако и, в целом, могут уволиться. Это потери со стороны HR-департамента, поэтому они должны эти моменты проговаривать, учитывать и соответствующе обрабатывать, но вся идея в том, что со стороны HR-департамента должны быть определённые задачи, приоритезированные в сторону увеличения, которые максимально подготавливают как себя, сам департамент, так и весь штат, весь персонал компании, для успешного перехода в облако. Если, грубо говоря, люди не будут готовы к переходу в облако, то ничего не произойдёт. Поэтому, с точки зрения людей, специалистов, нужна проработка и подготовка к этому переходу.

Следующее это угол обзора со стороны управления, руководства. Выделяют четыре подпункта, которые вы можете видеть на слайде. Идея в том, что высшее руководство должно учитывать процесс перехода в облако и внести соответствующие правки в управление портфолио, управление программами и проектами внутри компании, которые на текущий момент есть. Учитывать процесс перехода при расчёте бизнес метрик, метрик производительности и соответствующе делать выводы, и, в целом, готовиться к этому процессу, учитывать выгоды, которые они получают в своих планах.

Мы переходим к следующей части, это углы обзора к процессу перехода со стороны технологии, технические моменты. В нашем случае, первое – это Platform perspective, в котором есть шесть подпунктов. Идея в том, что IT департамент уже в процессе перехода должен понимать как у них будет происходить запрос новых вычислительных мощностей, как будет организована сеть, как будет организовано хранение базы данных и вот, в целом, изменится ли архитектура в облаке. Плюс нужно учитывать особенности самого приложения, которое вы разрабатываете. Некоторые моменты можно намного улучшить в облаке, некоторые моменты просто так взять и перенести не получится, поэтому эти моменты надо учитывать. Этот процесс не так вот в один день, на следующий день вы на работу пришли и все, вы все в облаке. На самом деле, этот процесс может затягиваться годами. Поэтому IT департамент при планировании будущих задач, или задач, которые в процессе, должны учитывать, что они будут построены и, в итоге, перенесены, либо сразу в облаке разработаны, и должны учитывать эти моменты и не делать такой функционал, который потом самим же создаст проблемы при переходе в облако.

Следующий пункт, со стороны технологии – это Security, безопасность, когда мы говорим про безопасность выделяют пять подпунктов, т.е. отдел техподдержки. Отдел администрирования IT-инфраструктурой должны проработать моменты с точки зрения безопасности и определить, как будут выдаваться доступы, доступы к пользователям, группам, приложениям в облаке, как будет происходить определение каких-то угроз, которые могут поступать, как будет обеспечиваться безопасность инфраструктуры, как будет происходить защита данных и как, в целом, будет происходить процесс обработки инцидентов, т.е. инциденты – это неотъемлемая часть жизненного цикла любого IT продукта, проблемы всегда возникают и задача компании – проработать этот процесс максимально гладко, для того чтобы инциденты быстро находили тех специалистов, которые должны разрешить этот инцидент и они успешно выполнялись и исправлялись.

Следующий и последний пункт с точки зрения технологии – это Operations, т.е. операционная деятельность, то что происходит каждый раз, каждый день. Сюда входят семь различных подпунктов. Первое – это как будет происходить в облаке мониторинг, мониторинг вашей IT-инфраструктуры, мониторинг приложений, которые есть в этой инфраструктуре, как происходит учёт IT ресурсов, которые у вас есть, как учитываются изменения этих IT ресурсов, как будет происходить отчётность и аналитика, с точки зрения IT-инфраструктуры и, в целом, этих инструментов. Также следует определить IT сервис каталог, это те сервисы в облаке, которые допустимы для использования внутри вашей компании, почему, в каком объёме и так далее, эти моменты тоже должны быть проработаны.

На этом мы завершаем нашу последнюю секцию. Давайте остановимся на основных моментах. Это то, что есть в облаке у AWS специально подготовленный документ, который позволяет вам учитывать все те проблемные моменты, которые могут возникнуть при переходе в облако. Выделяют два направления, это с точки зрения бизнеса и технологии, каждая из них состоит из трёх частей. Мы, в итоге, разобрали шесть углов обзора на процесс перехода более подробно, что туда входит и что должно быть сделано, фактически с каждого департамента, не только IT департамента внутри компании, для того, чтобы успешно перейти в облако. Ещё раз напоминаю, что именно эта тема AWS Cloud Adoption Framework предоставляется вам в ознакомительном виде, вы должны понимать, что такое, в целом, AWS Cloud Adoption Framework. Может простейший вопрос прийти на AWS экзамене, но более углублённые вопросы вот в этом направлении не придут. Поэтому вы теперь знаете, что есть такой документ, в будущем, когда вы начнёте работать как облачный инженер, впоследствии будете расти по карьерной лестнице и когда станете либо старшим специалистом, IT архитектором, может быть, техническим директором и вам понадобится перейти в облако, то это тот документ, который вам поможет. Этот документ на самом деле достаточно большой, объёмный. Вы его полностью проработаете, обработаете и соответственно всей компанией вместе будете двигаться для того, чтобы успешно перейти в облако.

На этом мы завершаем нашу сессию. Давайте пройдёмся по самым основным моментам. Первое – это то, что мы разобрали какие бывают вычислительные модели, какие бывают модели предоставления облачного сервиса. Дальше, мы разобрали 6 преимуществ облачных технологий. После этого мы определили, что такое AWS, какие у него есть сервисы, какие бывают группы сервисов. В самом конце, в четвёртой секции, мы разобрали что такое AWS Cloud Adoption Framework, т.е. это тот документ, который всячески помогает при переходе с локальной IT-инфраструктуры в облако. Здесь представлены все ссылки, которые вам могут понадобиться. Вы можете получить дополнительную более подробную информацию. На этом я завершаю эту сессию. Большое вам спасибо, надеюсь, вы сегодня узнали что-то новое и вы стали ещё ближе к успешной сдаче AWS экзамена. Увидимся на следующих занятиях. До свидания!