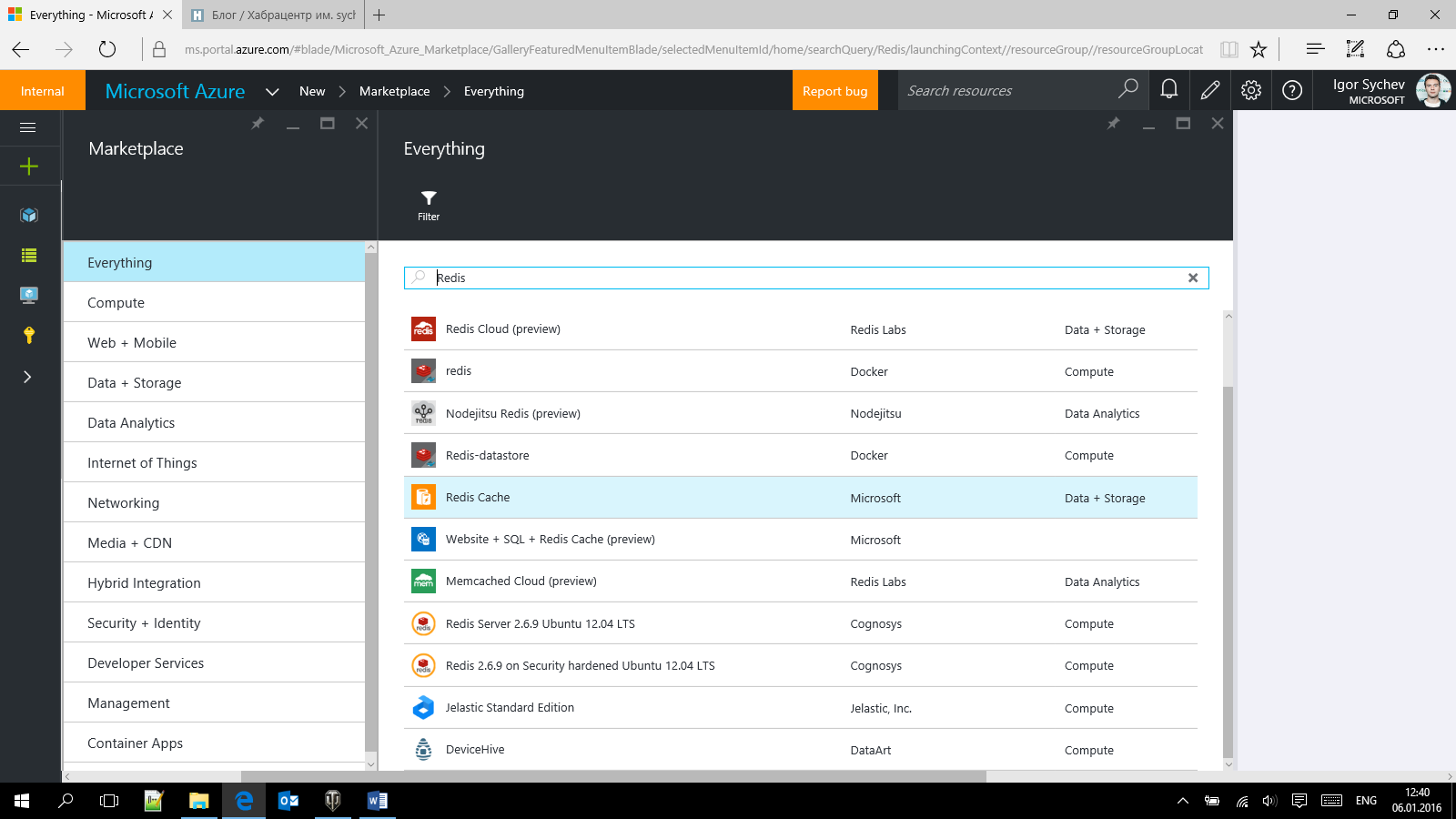
В начале 2015 года я уже рассказывал <http://habrahabr.ru/company/microsoft/blog/247531/> , что Microsoft теперь рекомендует использовать Redis в качестве кэша для всех новых проектов в Azure, а объявлена эта рекомендация была еще в 2014 году.

В ноябре 2015 нам сообщили, что кэш внутри cloud service признан устаревшим <https://azure.microsoft.com/en-us/documentation/articles/cache-faq/#which-azure-cache-offering-is-right-for-me>, и в будущих версиях sdk мы удалим работу с ним, и надо использовать Redis Cache. Именно это событие и побудило меня рассказать, что добавилось за прошедший год в Redis.

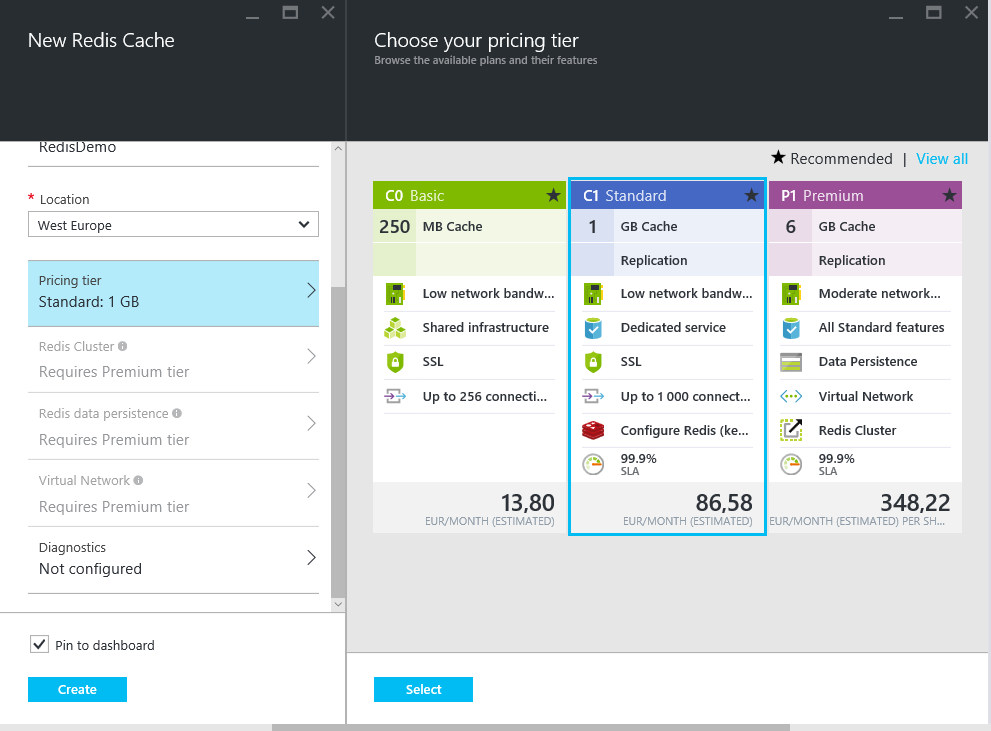
Давайте пойдем не от документации, а от портала.

Момент первый: есть много разных решений на портале, которые в названии имеют слово Redis. Это наши партнеры постарались, за что им спасибо. Нам же нужно то, которое опубликовала Microsoft.



Открываем панель настроек

Первый и самый базовый вопрос: какой набор фич выбрать (от этого набора фич зависит и стоимость)?



Не думайте, что на картинке все возможные варианты с объем памяти, он может быть больше. Тут показаны именно наборы фичей.

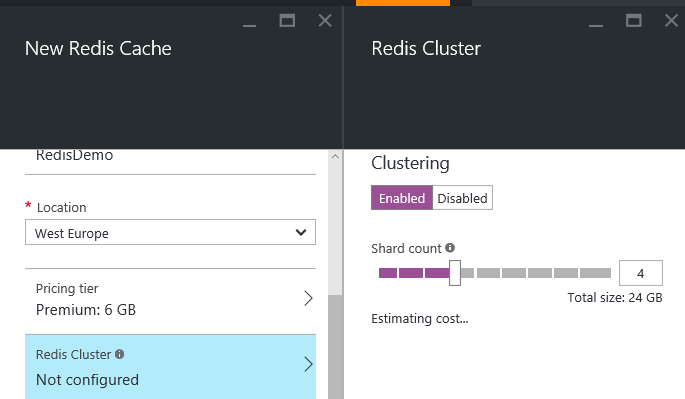
Какие варианты мы видим?

* **Basic -** -инфраструктура будет не выделенная конкретно под вас, а общая еще с кем-нибудь.
* **Standard** - ресурсы уже будут зарезервированы под вашу базу - и никаких шумных соседей, появляется SLA, т.к. и объем базы выше, и появляется вторая реплика ваших данных.
* **Premium** - самая крутая (включает все опции от standard и еще много чего… но и самая дорогая), которую я предлагаю рассмотреть ниже.

Давайте разберем каждую фичу по отдельности.

**Clustering <https://azure.microsoft.com/en-us/documentation/articles/cache-how-to-premium-clustering/>**

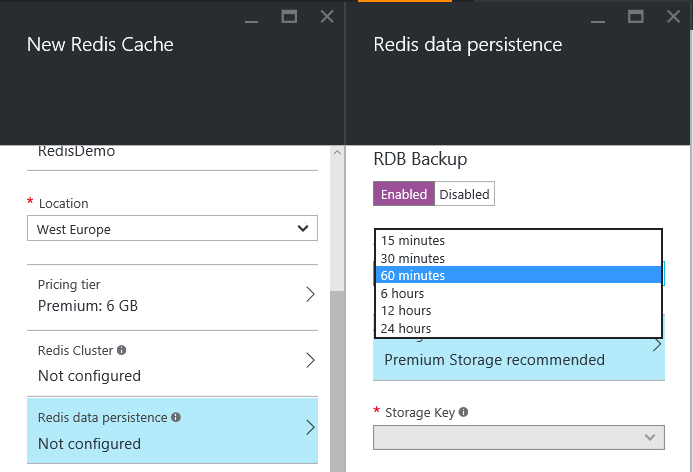
Суть кластера - разбить данные на части и получить горизонтально-масштабируемую конфигурацию. Redis для вас раскидает ключи по нескольким шардам. Год назад в azure, cluster создать было нельзя, да и вне azure эта фича была “alpha quality” (требующая доработок).



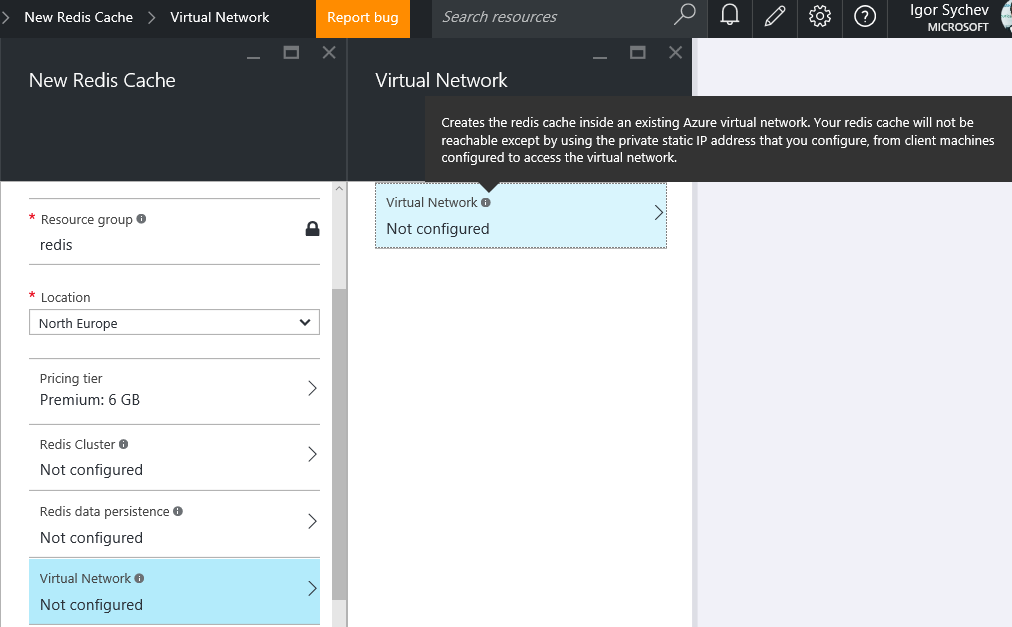
Как мы видим, чем больше шардов мы используем, тем выше максимальный (суммарный) размер хранилища. Правда, чем больше объем, тем выше будет цена за итоговое решение. Максимально можно собрать кластер на 530 Гб (10 шардов по 53 Гб)

**Persistence** [**https://azure.microsoft.com/en-us/documentation/articles/cache-how-to-premium-persistence/**](https://azure.microsoft.com/en-us/documentation/articles/cache-how-to-premium-persistence/)

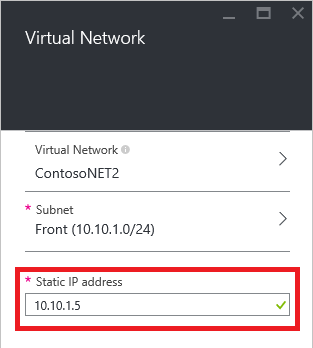
Persistence - это сохранение inmemory базы данных на диск или - если быть точнее - на storage. Сохранение на данный момент происходит единым куском, а не инкрементально, т.к. это RDB <http://redis.io/topics/persistence> способ. В планах есть и второй способ сохранения данных в storage, AOF <https://azure.microsoft.com/en-us/blog/azure-redis-cache-public-preview-of-premium-tier/> , но когда его сделают доступным информации нет.

Мы рекомендуем использовать premium storage, потому что он быстрее, т.к. основан на SSD. Все-таки писать из быстрой inmemory-базы на медленное hdd-хранилище – это достаточно редкое сочетание. Но если Вы хотите – пожалуйста, пишите на standard storage.

**Интеграция с виртуальными сетями** [**https://azure.microsoft.com/en-us/documentation/articles/cache-how-to-premium-vnet/**](https://azure.microsoft.com/en-us/documentation/articles/cache-how-to-premium-vnet/)



Я специально оставил подсказку с портала, чтобы было проще объяснить. Вы можете обращаться к созданному для Вас Redis-кластеру либо из интернета, либо из виртуальной сети. Либо по public ip, либо по private. Лично мне этой функции ранее больше всего не хватало.

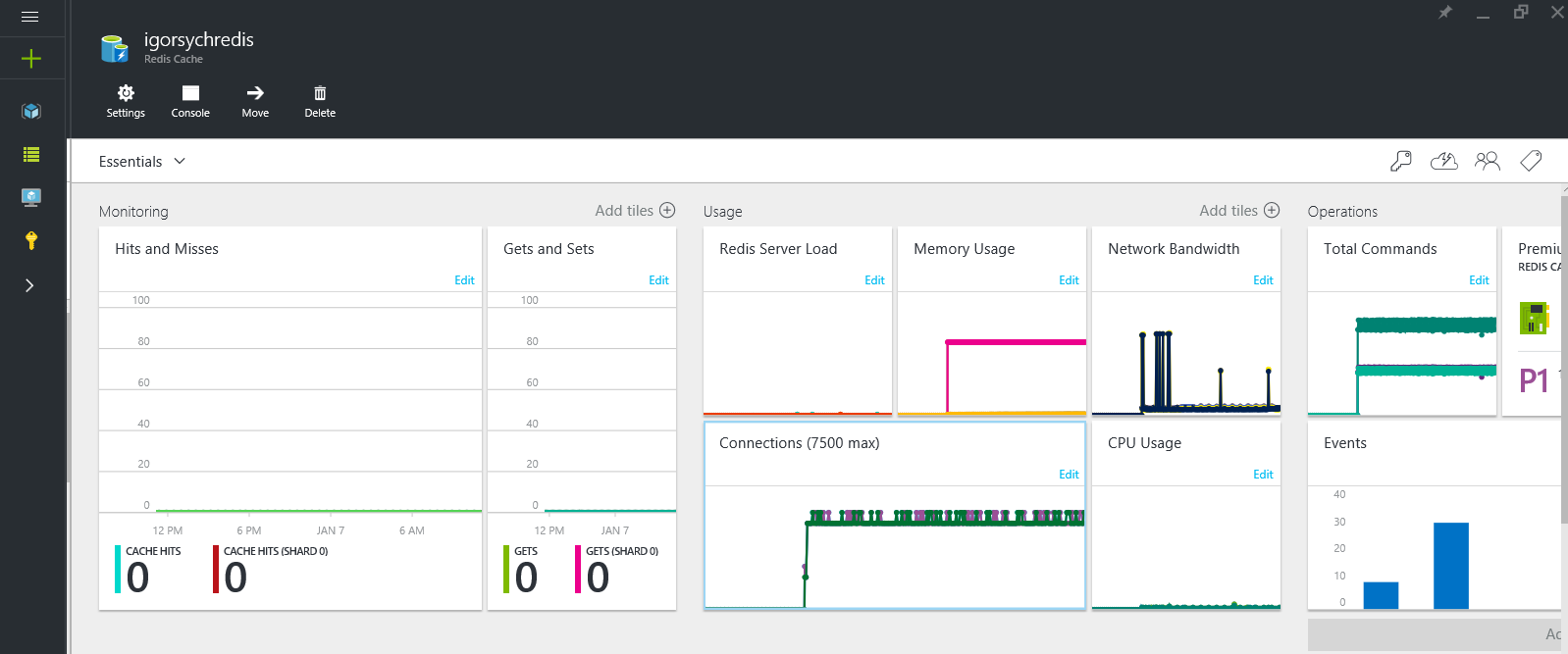


Есть и второе ограничение: на данный момент мы можем использовать “Classic” виртуальные сети, что лично для меня печально. Если вы с этим согласны, прошу проголосовать за мою идею, чтобы команда разработки приоритезировала эту задачу <https://feedback.azure.com/forums/169382-cache/suggestions/11327502-add-integration-with-v2-virtual-networks>.

**После того, как мы создали наш redis cache, можно посмотреть на остальные его опции.**

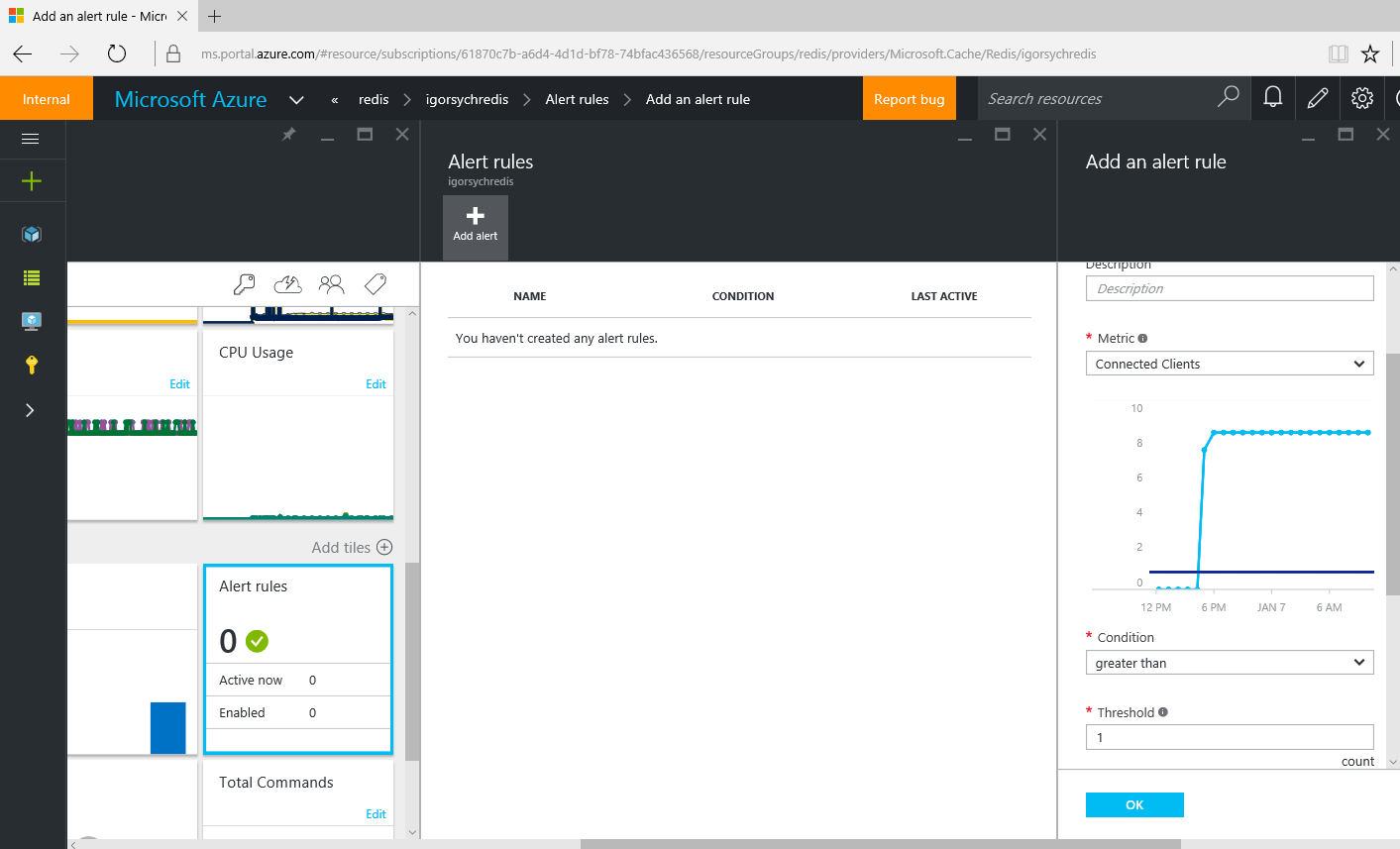
Во-первых, все указанные нами ранее параметры можно изменить: не хватает объем=> увеличиваем объем, решили добавить еще 1 шард=> устанавливаем sharding в enable, добавляем шард, а redis перераспределяет ключи (мы должны понимать, что это не мгновенная операция).

**Мониторинг**



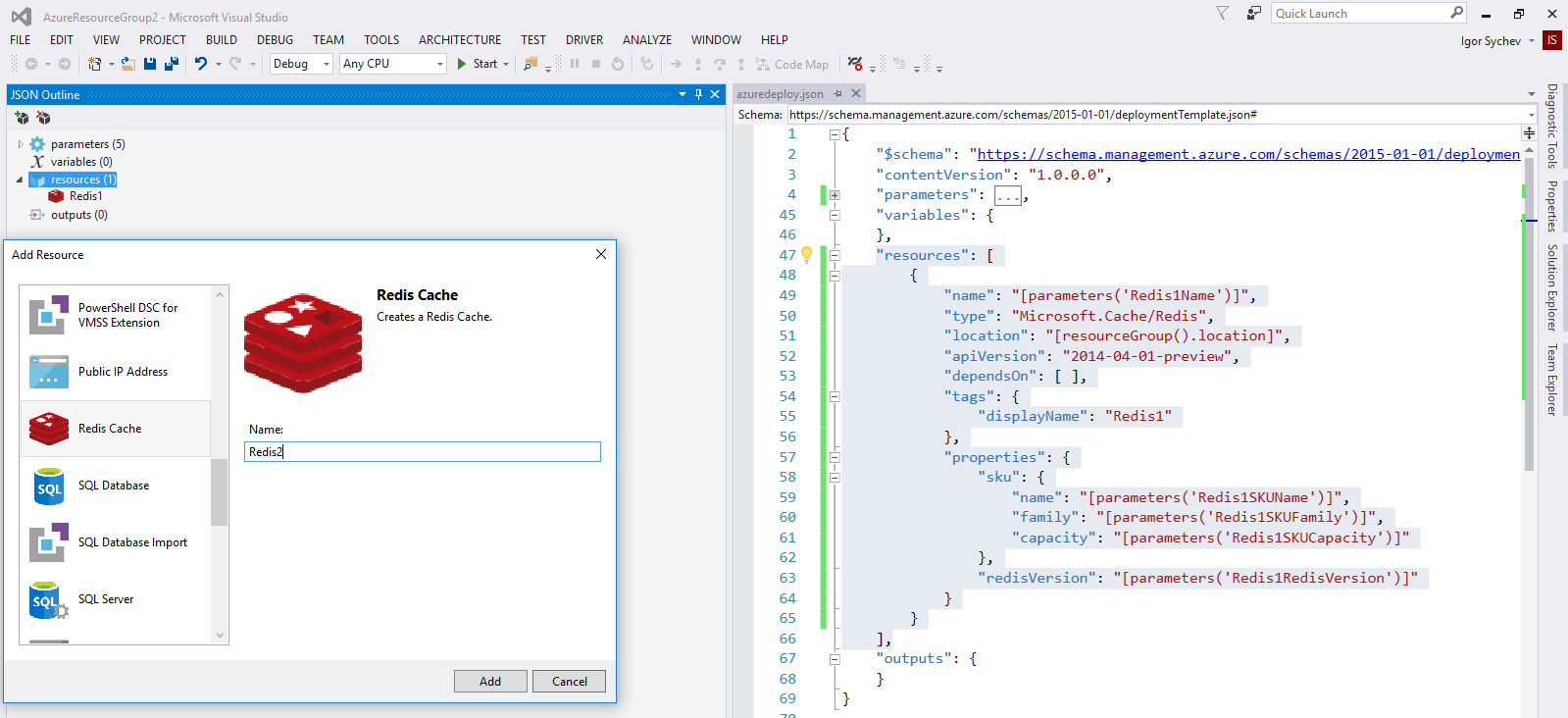
Если вы включили опцию мониторинг, то эксплуатация становится намного проще: вы будете знать и сколько у вас активных коннектов было, и размер потребляемой памяти, и еще множество метрик, которые можно настроить.

Ну и естественно, вы можете настроить нотификации (alerts), если какие-то метрики вышли из диапазона допустимых значений. В частности, можно указать количество соединений, и, если их число превысит порог, получить сообщение на почту.



**Configuration as Code**

Конфигурацию Redis можно задать в виде ARM шаблона <https://azure.microsoft.com/en-us/documentation/articles/cache-redis-cache-arm-provision/>

В Visual Studio это будет выглядеть как JSON файл. Считайте, что это design time шаблон. 

А также мы можем посмотреть на execution time представление этого шаблона по адресу: <https://resources.azure.com/> 