Azure API Apps Part III

CSCI E-94

Fundamentals of Cloud Computing - Azure

Joseph Ficara

Agenda

- Validation
- Azure SQL Basics
- Entity Framework Core Basics
- Strongly Typed Settings

Validation

- Validation requires three steps
  - 1. Annotate the model with Data Annotations
  - 2. Check ModelState.IsValid
    - In action methods
  - 3. Return errors via BadRequest(ModelState)

Validation

- Built in Data Annotations
  - See: http://bit.ly/2SbCUJd
  - Compare
  - CreditCard
  - CustomValidation
  - MaxLength
  - MinLength
  - Range
  - Url
  - RegularExpression
  - Required
  - StringLength
  - Password
  - EmailAddress
  - FileExtensions
  - Phone

Validation

- Model Data Annotations can be applied to
  - A property - to validate that property
    - Independently of any other property
  - The model data as a whole
    - CustomValidation & ValidationAttribute
      - Either support multiple property validation
    - For example:
      - Verify
        - City and State are valid for the ZipCode
          - By examining the collection of items

Validation

Model property validation

```
// Name is required and has a max length of 30 characters
[Required]
[StringLength(30)]
public string Name { get; set; }

// Age must be between 21 and 120
[Range(21,120)]
public int Age { get; set; }
```

Validation

- To use the CustomValidation attribute
  - Annotate the Model class
    - With the CustomValidation attribute
      - Provide the
        - Model type as the first parameter
        - Validation method name as the second parameter
  - Implement the validation method
    - Its name is the second parameter value
      - The first parameter is the Model instance object
      - The second parameter is the ValidationContext
        - Typically, not used for class level validation

Validation

Multi-Property Model validation via CustomValidation

```
[CustomValidation(typeof(Customer),"ValidateNameAndEmail")]
public class Customer
{
   public static ValidationResult ValidateNameAndEmail(Customer customer,
                                                        ValidationContext ctx )
{
   // Verify that the email address contains either the first or last name
   string[] nameParts = customer.Name.ToLowerInvariant().Split(' ');
   ...
   if (customer.EmailAddress.ToLowerInvariant().Contains(nameParts[0]))
   {
      return ValidationResult.Success;
   }
   ...
   return new ValidationResult("Email must contain first or last name",
                               new List<string> { "Name","EmailAddress" });
}
```

Demo

Validation

`CustomerDemoSolution`

Validation

- Custom error messages are supported
  - Data Annotations allow for user defined
    - Hard coded error message
    - Resource based error messages
      - Supporting localized error messages
- Custom Data Annotations are supported
  - Derive from the ValidationAttributeClass
    - Override IsValid method
      - This is where your validation logic goes
    - Optionally override FormatErrorMessage

Validation

- Best Practice:
  - Design your own error response payload
  - Include both
    - Numeric representation of the specific error
      - So, client code can be written for it
    - Human readable description of the error
      - So, developer understands the issue
  - Ideally
    - Validate as much as you can and return ALL errors
    - Improves API usability

Validation

Example error response

```
public class ErrorResponse
{
   //The set of properties with invalid values
   public List<PropertyError> propertyErrors { get; set; }

   // Provides a human readable description of the error not localized
   public string topLevelErrorDescription { get; set; }

   // A numeric representation of the error that can be used to code against
   public int topLevelErrorNumber { get; set; }
}
```

Validation

Example error response …

```
// Defines the property error structure
public class PropertyError
{
   // The name of the property with invalid input
   public string name { get; set; }

   // The invalid input value of the property
   public string value { get; set; }

   // Provides a human readable description of the error not localized
   public string description { get; set; }

   // A numeric representation of the error that can be used to code against
   public int errorNumber { get; set; }
}
```

Validation

Azure SQL Basics

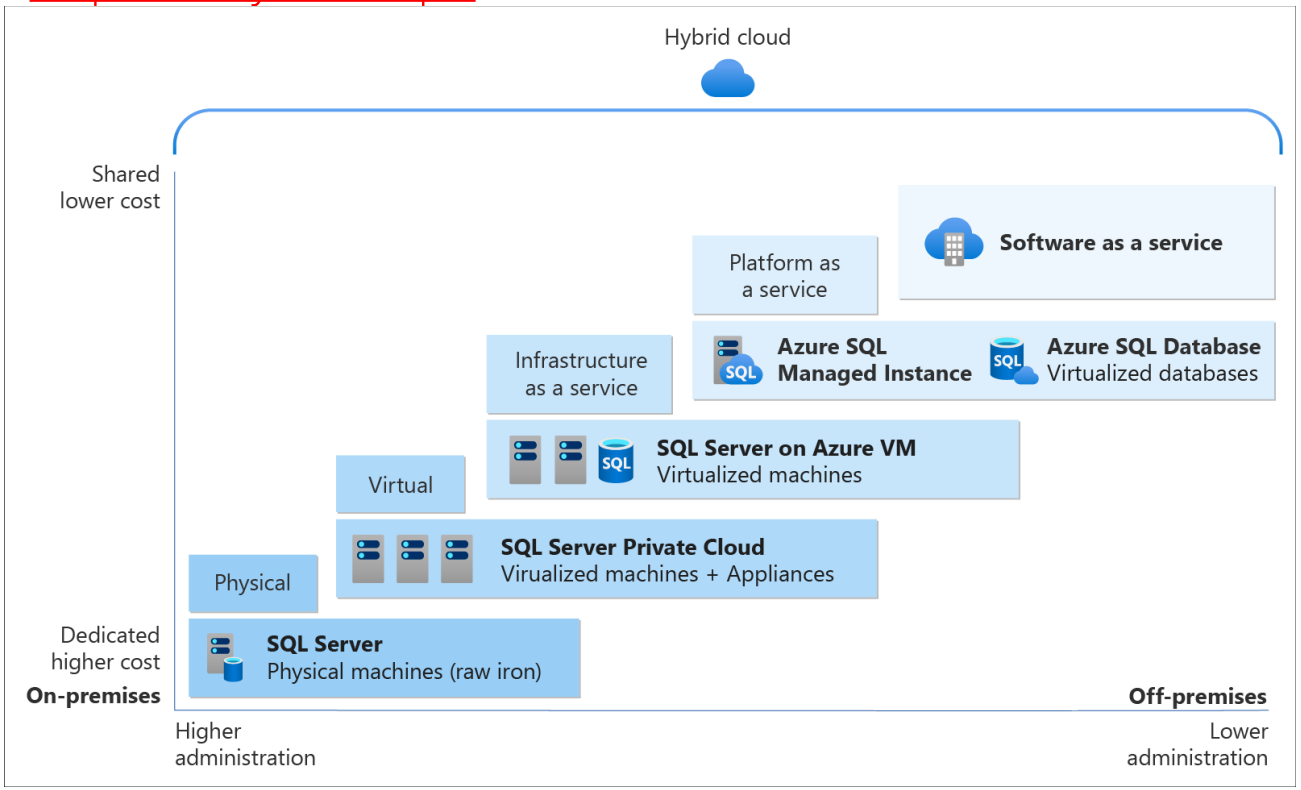
- What is Azure SQL?
  - Microsoft's PaaS fully managed SQL offering
- Why do you care?
  - All the benefits of PaaS
  - Migration path from on premise to cloud
    - Feature Comparison See: http://bit.ly/2FPaKuh
      - No SQL Jobs
      - No SQL CLR
      - Some on premise T-SQL not supported
      - No Extended Stored Procedures

Azure SQL Basics
Why do you care …

- Automated low/no touch features
  - Automatic backups
  - Click to Geo Replicate
  - Dynamically Scalable
    - Average of 4 seconds interruption
  - New features appear in Azure SQL first!
  - Business continuity
    - Point-In-Time Restore
    - Active geo-replication
- PaaS and IaaS comparison
  - See: http://bit.ly/2s8hq51

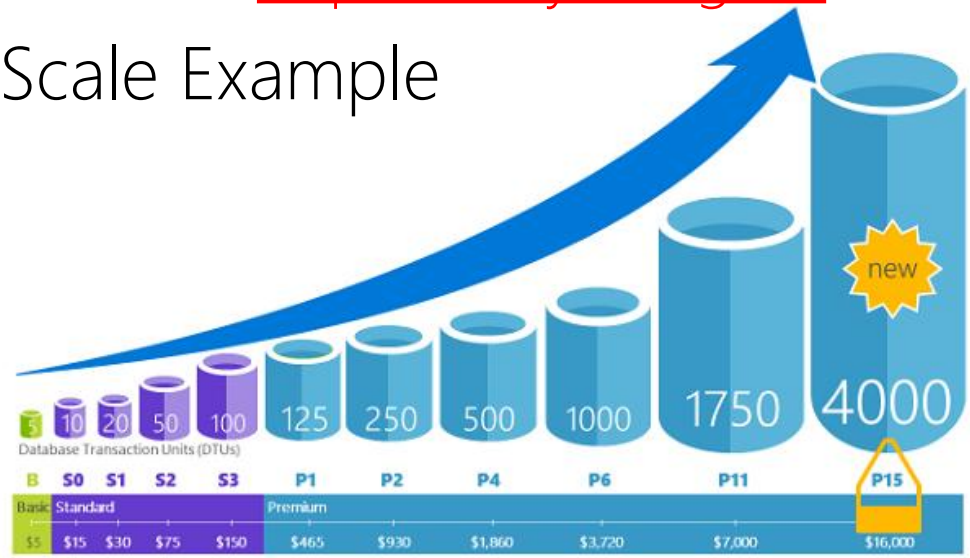
Azure SQL Basics

- Credit: Azure SQL Documentation
  http://bit.ly/2s8hq51



Azure SQL Basics

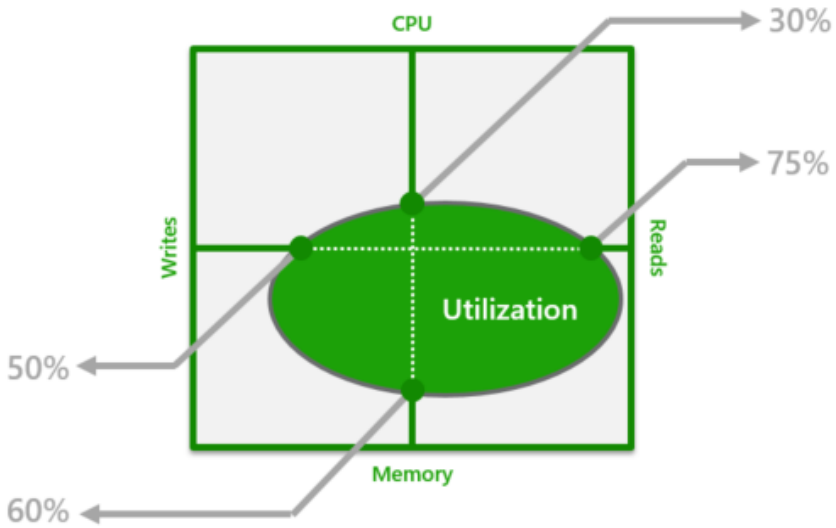
- Wide range of scalability
  - Single Database Scale: https://bit.ly/3uc1T2q
  - Hyperscale service tier: https://bit.ly/3ugvHed
  - Elastic Pool: https://bit.ly/3LdglNR
- DTU Scale Example



Credit Microsoft docs.microsoft.com

Azure SQL Basics

- What are DTUs?
  - Data Throughput Units
  - Relative resources assigned to the DB
  - Blended measure
    - CPU
    - Memory
    - Read/Write rates



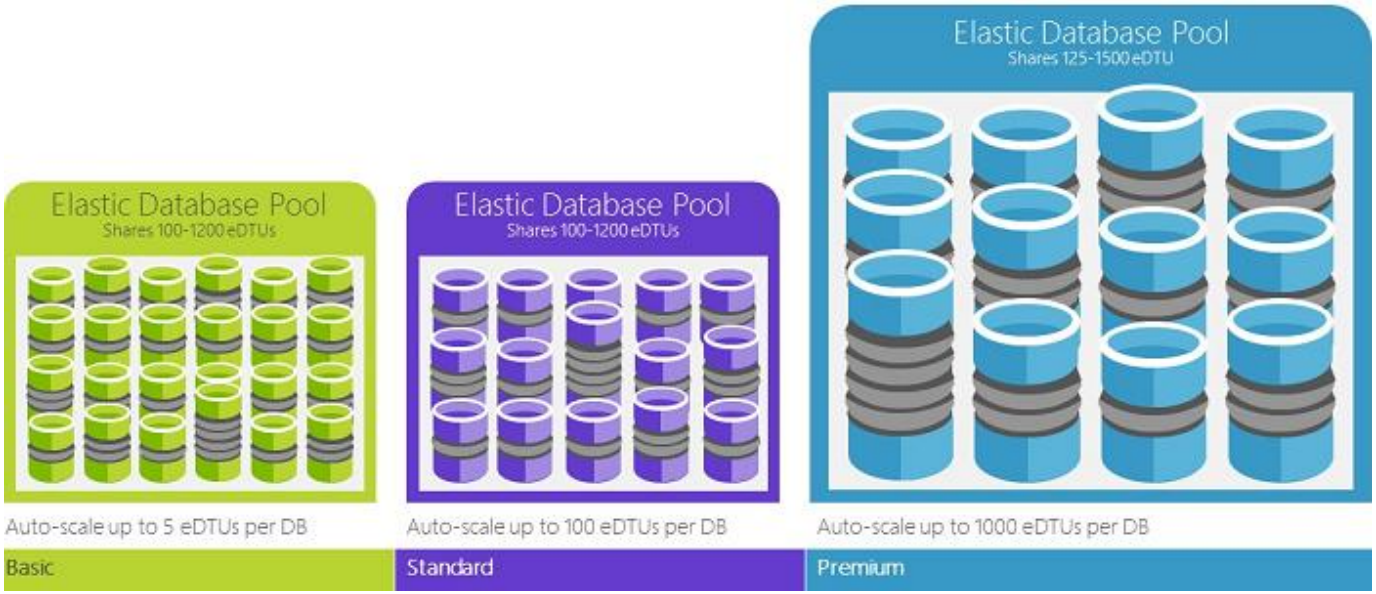
Credit Microsoft docs.microsoft.com

Azure SQL Basics
What are DTUs?...

- Doubling DTUs
  - Doubles set of resources available
- Example:
  - Basic has 5 DTUs
  - P15 has 4000 DTUs
    - Increase of 800x
      - More compute power than a Basic Db with 5 DTUs
  - See: http://bit.ly/2BGoHwS
- DTU Calculator: https://dtucalc.azurewebsites.net/

Azure SQL Basics

- Elastic pools maximize resource utilization
  - Credit: Azure SQL Documentation https://bit.ly/3AR1dAA



Credit Microsoft docs.microsoft.com

Azure SQL Basics

- Hyperscale service tier
  - 99.995% availability
  - Uses the vCore-based purchasing model
  - Capabilities
    - 100 TB database size
    - Nearly instantaneous database backups
      - With no IO impact on computing resources
    - Fast database restores in minutes
    - Rapid scale out & up

Azure SQL Basics

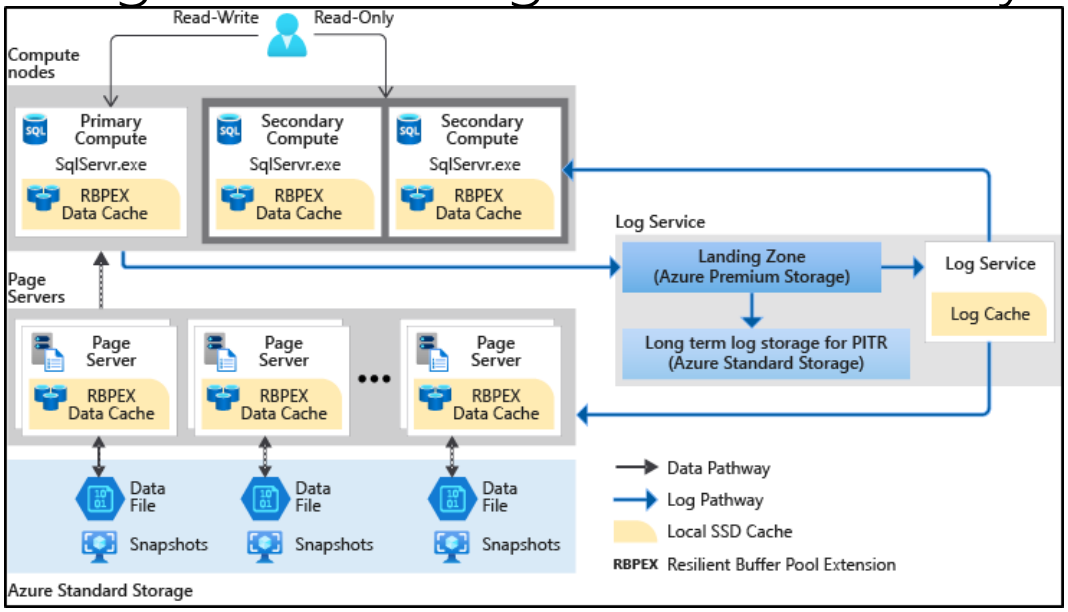
Hyperscale service tier

- When should you use it
  - Large database on-premises moving to cloud
  - Cloud limited by maximum database size
  - Small database that
    - Require fast vertical and horizontal scaling
      - Compute scaling
      - instant backup
      - fast database restore

Azure SQL Basics

Hyperscale service tier

- Separation of
  - Query processing engine
  - Long-term storage and durability of data



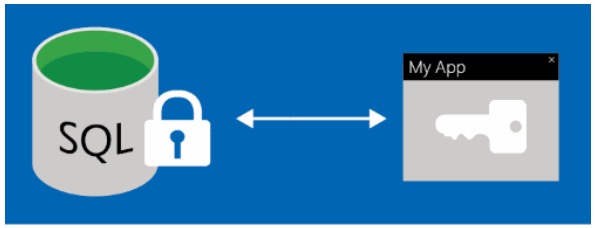
Credit Microsoft docs.microsoft.com

Azure SQL Basics

- Automatic performance monitoring
  - Suggests indexes based on production usage
    - Can auto create and remove
  - Recommendation:
    - Review before applying
- Intelligent threat detection
  - Alerts upon suspicious activities
- Auditing
  - Track database events and writes

Azure SQL Basics

- Data encryption at rest
  - **T**ransparent **D**ata **E**ncryption (TDE)
- Data encryption in motion
  - Always Encrypted
    - See: http://bit.ly/2EHqGQ7



- Dynamic data masking
  - Policy controlled hides sensitive data
    - See: http://bit.ly/2vZQo1T

Azure SQL Basics

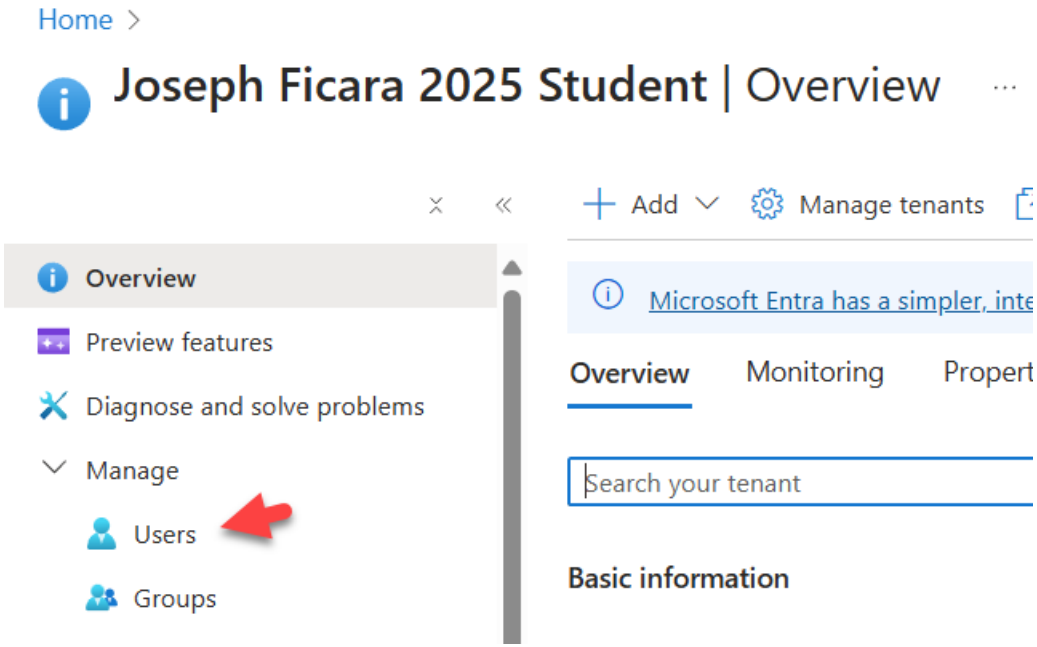
- Row-level security (RLS)
  - Control access to rows based on
    - Group membership, Execution context
    - See: http://bit.ly/2sc303x

Azure SQL Basics

- Let's do a walkthrough
  - Creating Azure SQL instance
- Basic Steps:
  - Create an Entra User **dbadmin**
    - To choose later as the admin to your db server
  - Choose Azure SQL
  - Pick Single Database
  - Enter database name
  - Create / Pick resource group
  - Create a server
  - Choose Basic 5 DTUs, 100MB ($4.90 usd)

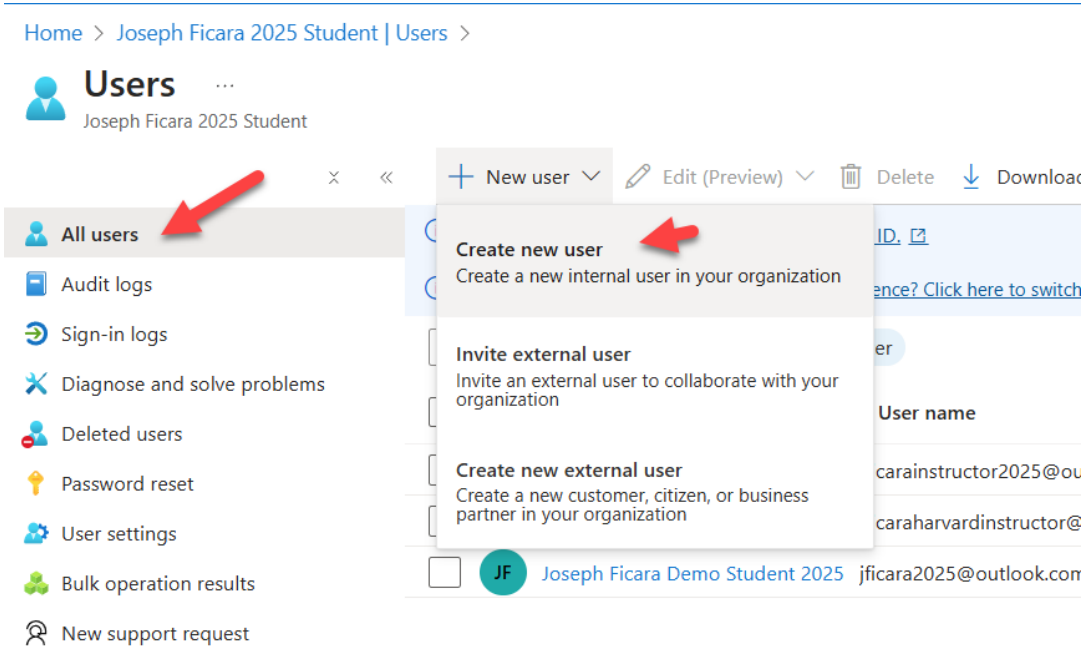
Azure SQL Basics

- Go to Entra ID
  - Select Users



Azure SQL Basics

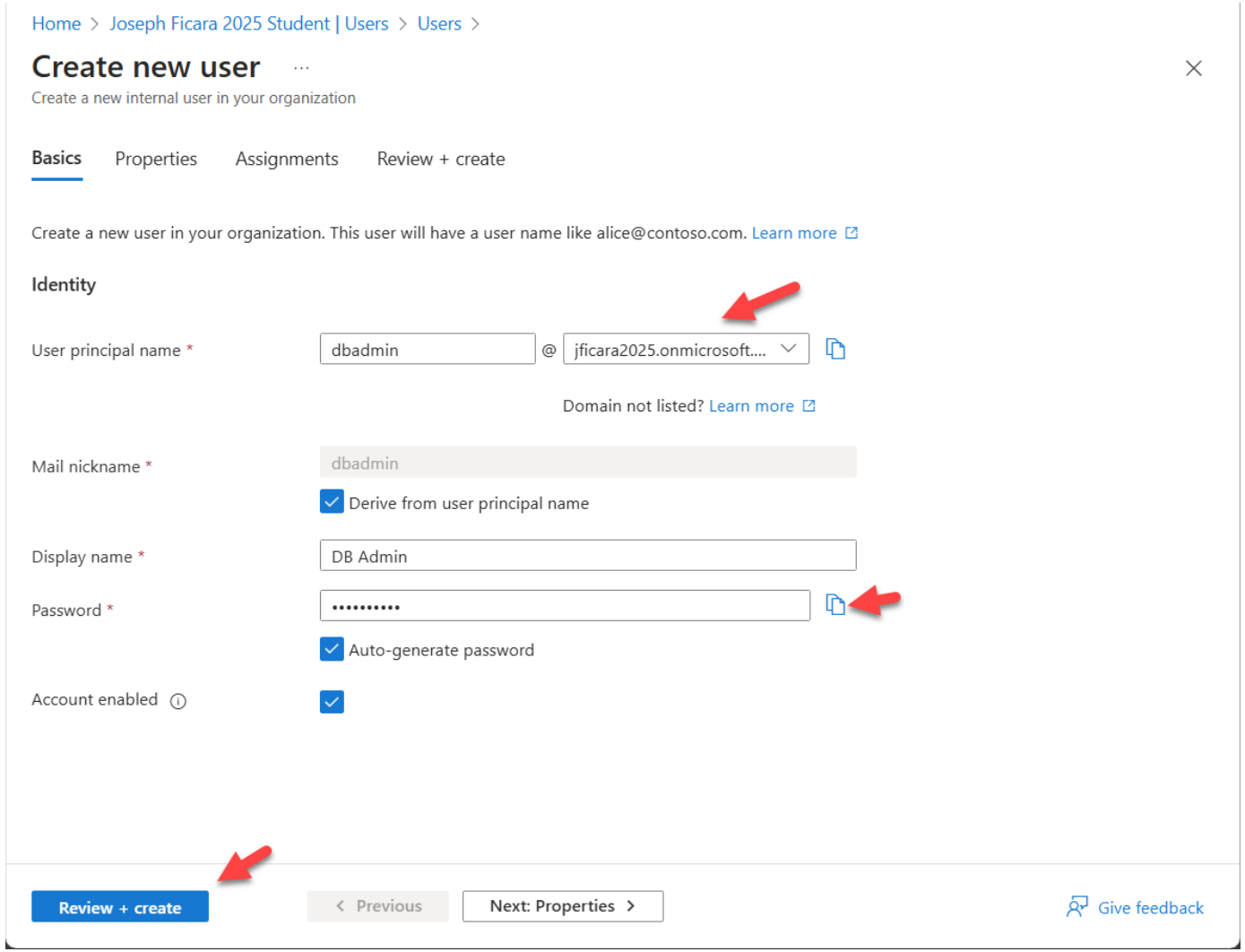
- Select All Users
  - Click New user
    - Create new user



Azure SQL Basics

- Enter the user's details
  - User Principal name **dbadmin**
  - Display name: DB Admin
    - Password - Copy it!
    - Auto-generate password
      - Is checked
    - Account enabled
      - Is checked

Azure SQL Basics

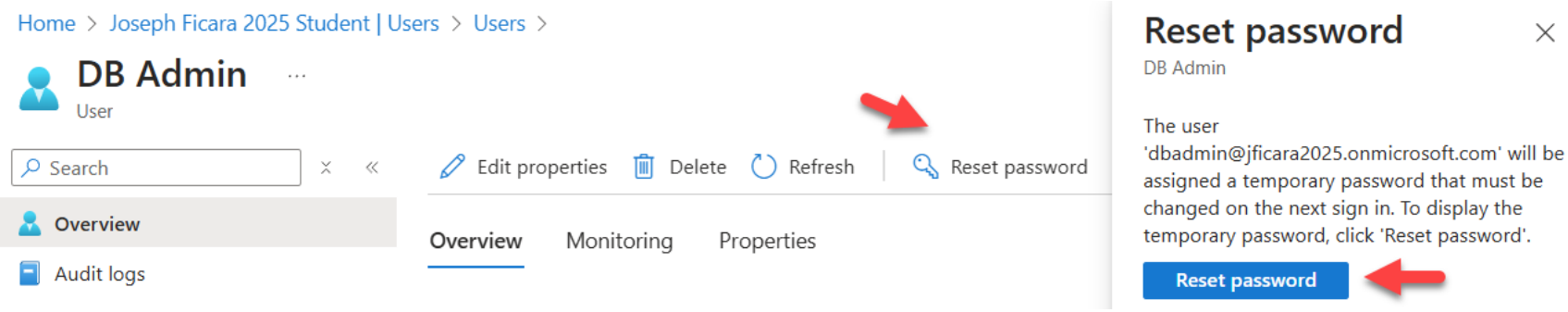


Azure SQL Basics

- Login to the portal
  - User Principal name **dbadmin**
    - dbadmin@yourdomain.onmicrosoft.com
  - Use the password you copied earlier
    - If you forgot to do this
      - Go to the user in Entra Id
      - Assign a new password
  - Assign a new password
  - Add multi factor authentication support
    - **Log in to portal, set new password, setup MFA**

Azure SQL Basics

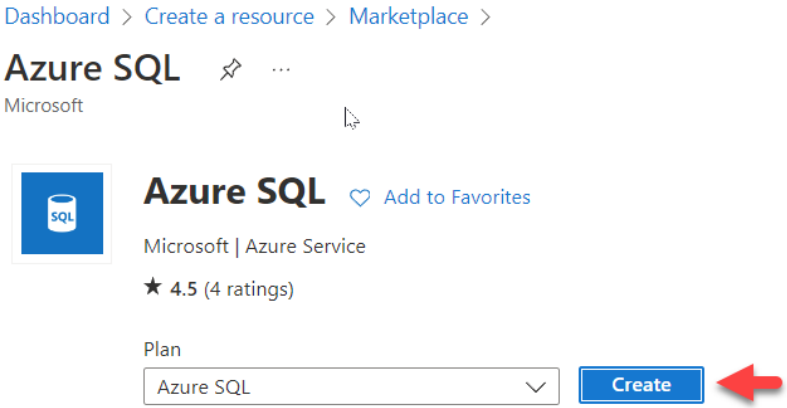
- If you forgot the password
  - To your Entra Id User
    - Go to Entra Id
    - Select the User
    - Reset the password



Azure SQL Basics

- Basic Steps:
  - DONE: Create an Entra User **dbadmin**
    - To select later as the admin to your db server
  - Create a SQL Database
    - Choose Azure SQL
  - Pick Single Database
  - Enter database name
  - Create / Pick resource group
  - Create a server
  - Choose **Basic 5 DTUs, 100MB ($4.90 usd)**

Azure SQL Basics



Azure SQL Basics

Dashboard > New > Azure SQL >

Select SQL deployment option

Microsoft

Feedback

How do you plan to use the service?

SQL databases

Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type

Single database

Create | Show details

SQL managed instances

Best for most migrations to the cloud. Lift-and-shift ready.

Resource type

Single instance

Create | Show details

SQL virtual machines

Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image

Create | Show details

Single database

Single databases are a great fit for modern, cloud-born applications that need a fully managed database with predictable performance.

Featured capabilities:

- Hyperscale storage (up to 100TB)
- Serverless compute
- Easy management

Elastic pool

Elastic pools provide a cost-effective solution for managing the performance of multiple databases with variable usage patterns.

Featured capabilities:

- Resource sharing for cost optimization
- Simplified performance management

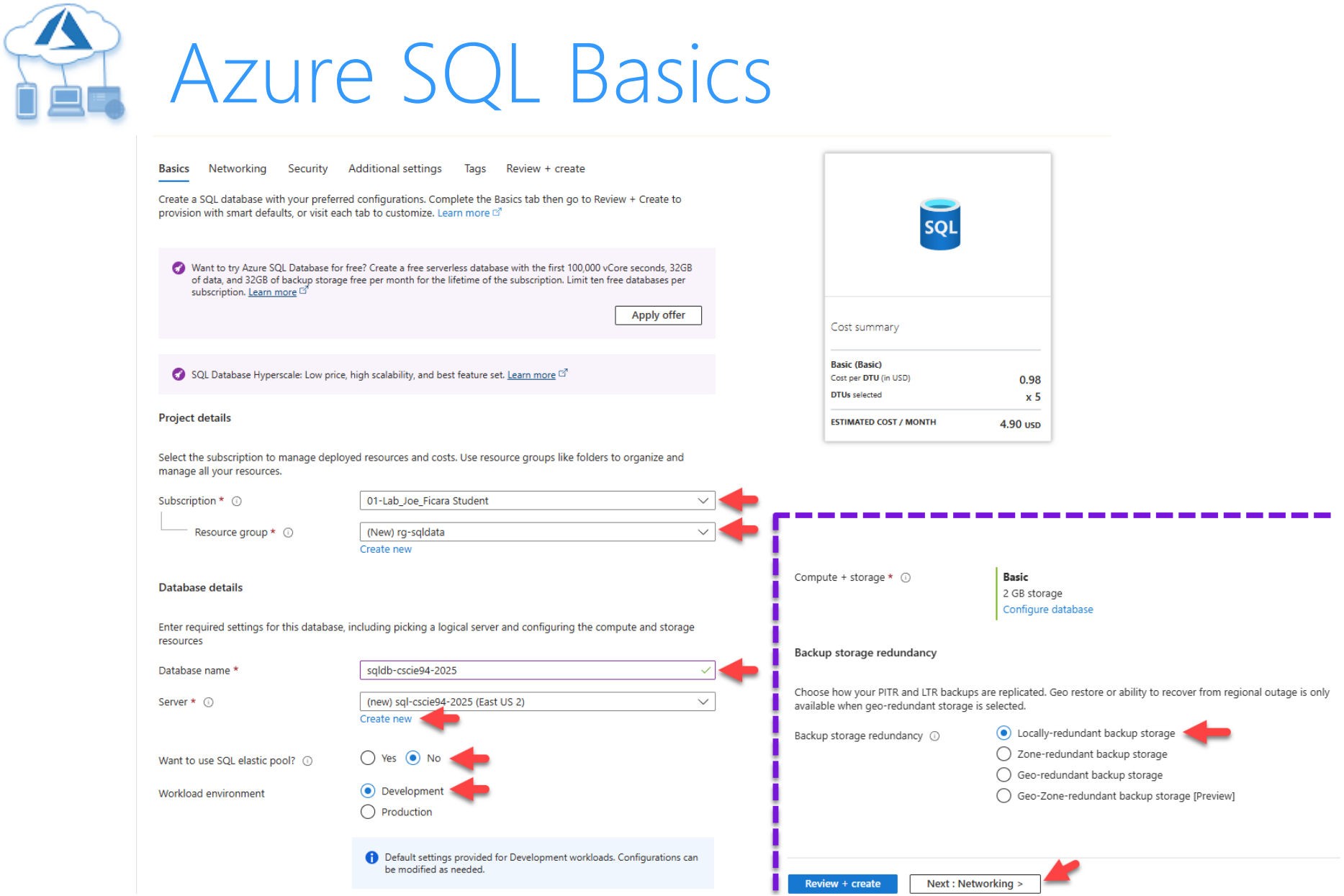
Database server

Database servers are used to manage groups of single databases and elastic pools.

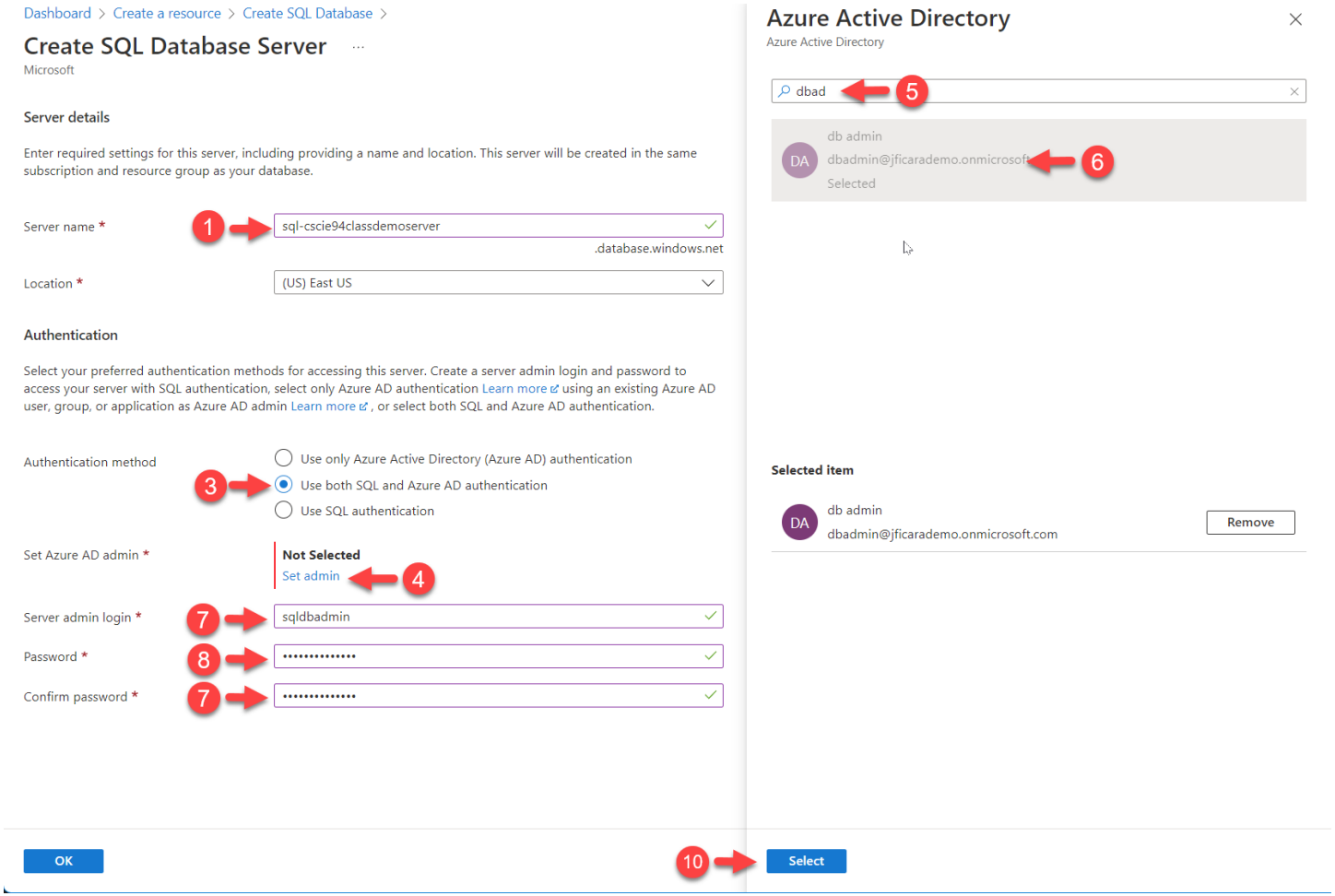
Featured capabilities:

- Access management
- Backup management
- Business continuity management

Azure SQL Basics



Azure SQL Basics



Azure SQL Basics

Dashboard > Create a resource > Marketplace > Azure SQL > Select SQL deployment option > Create SQL Database >

Configure ...

Feedback

Service and compute tier

Select from the available tiers based on the needs of your workload. The vCore model provides a wide range of configuration controls and offers Hyperscale and Serverless to automatically scale your database based on your workload needs. Alternately, the DTU model provides set price/performance packages to choose from for easy configuration. Learn more

Service tier: Basic (For less demanding workloads)

Compare service tiers

DTUs Compare DTU options

5 (Basic)

Data max size (GB): 2

Cost summary

Basic (Basic)

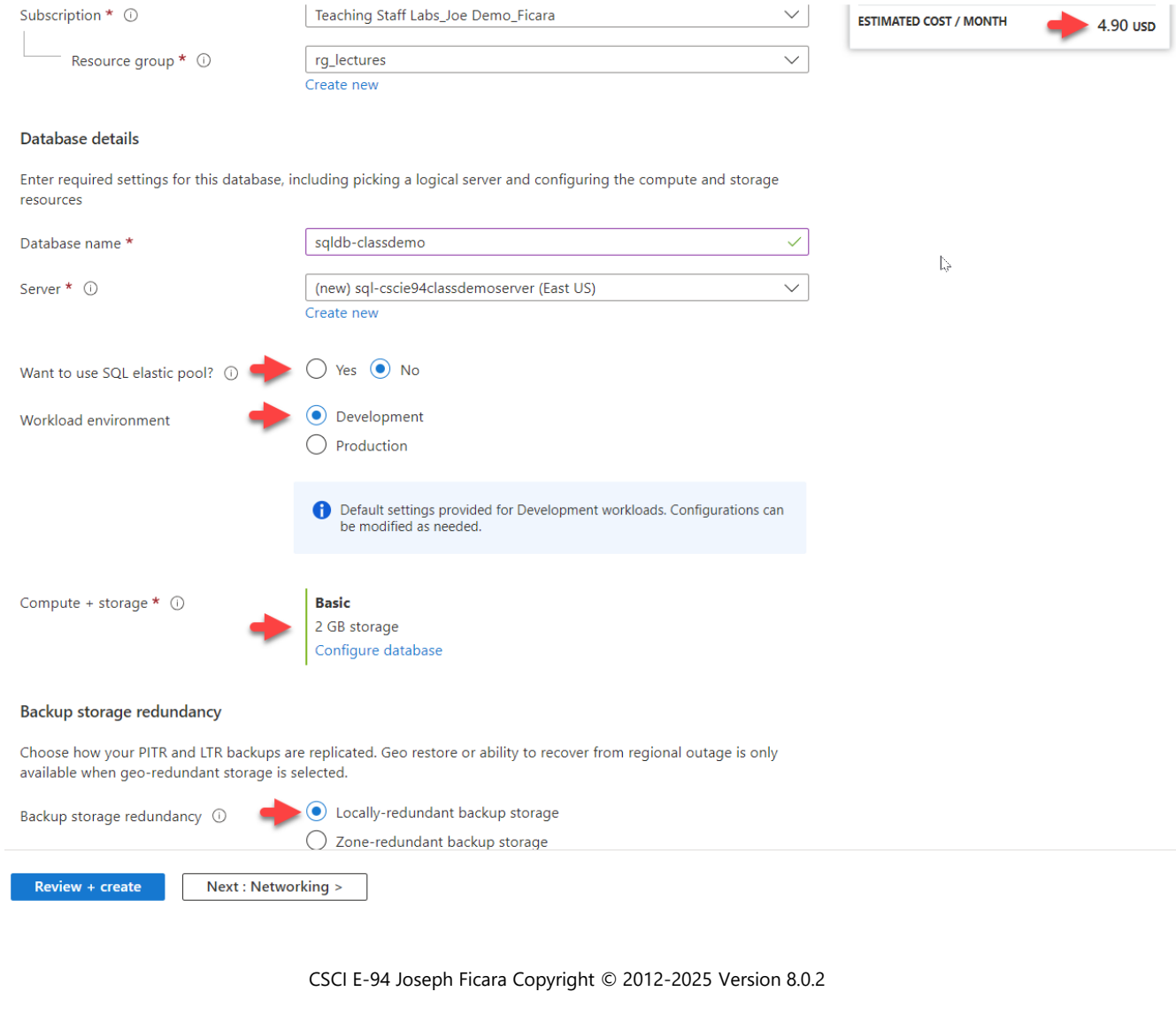
Cost per **DTU** (in USD) 0.98

DTUs selected x 5

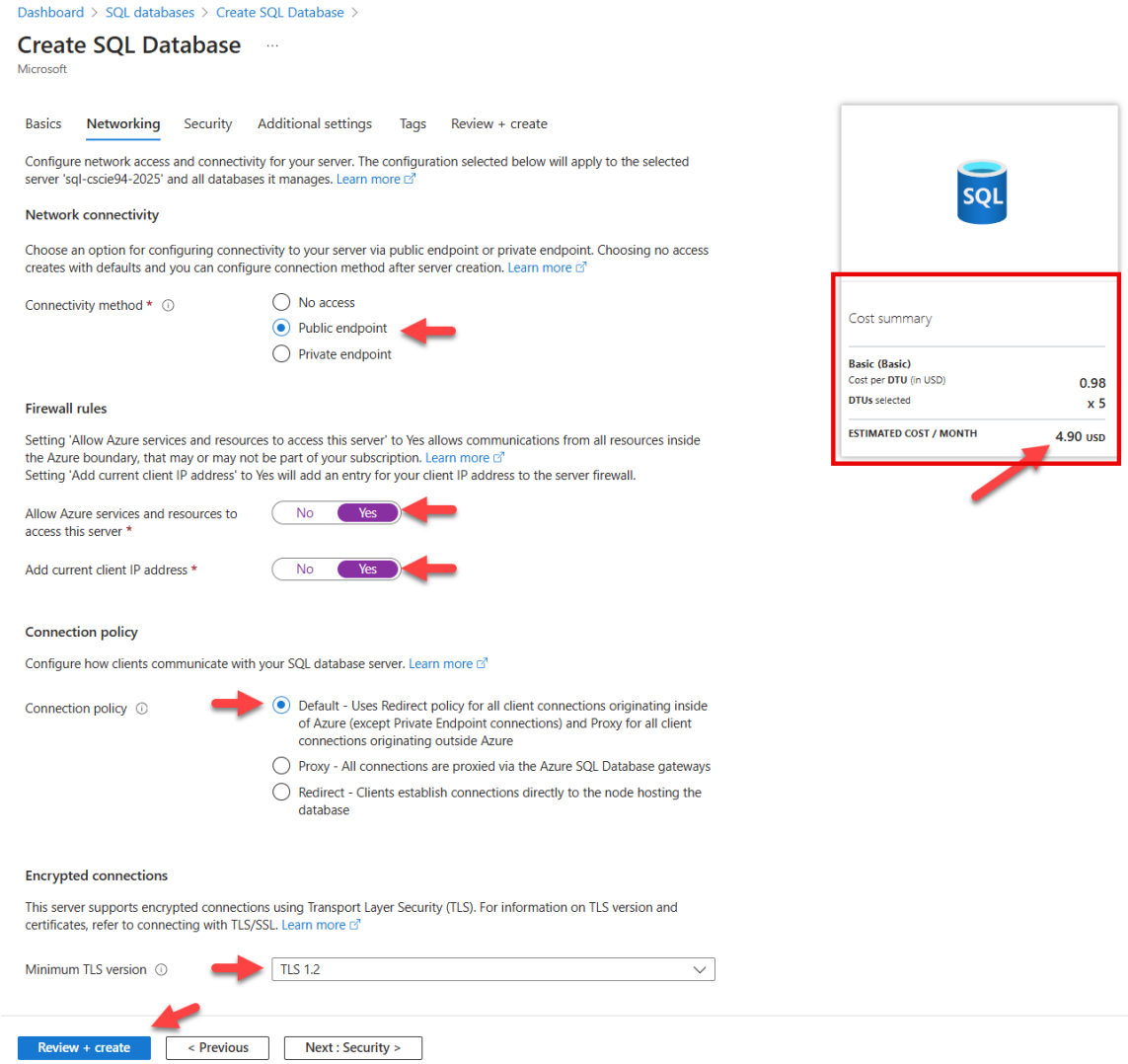
ESTIMATED COST / MONTH 4.90 USD

Apply

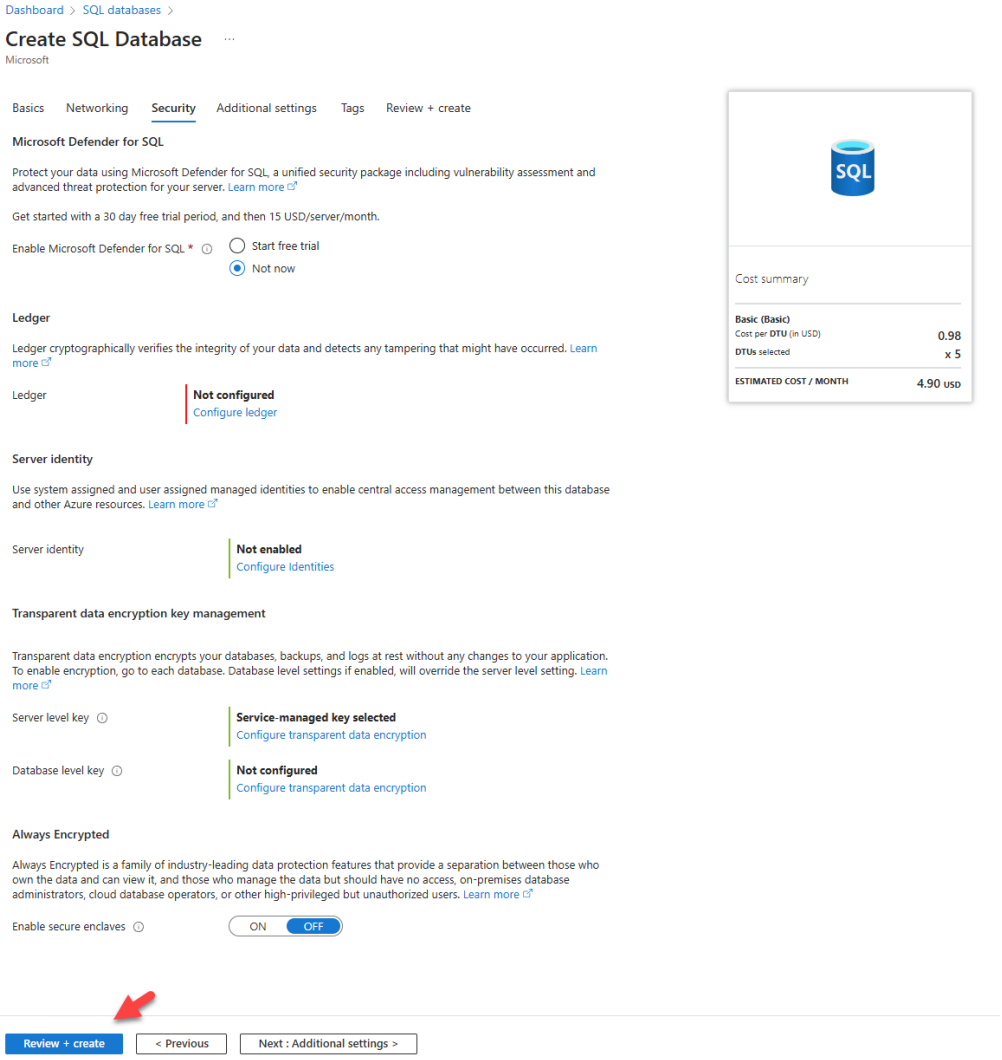
Azure SQL Basics



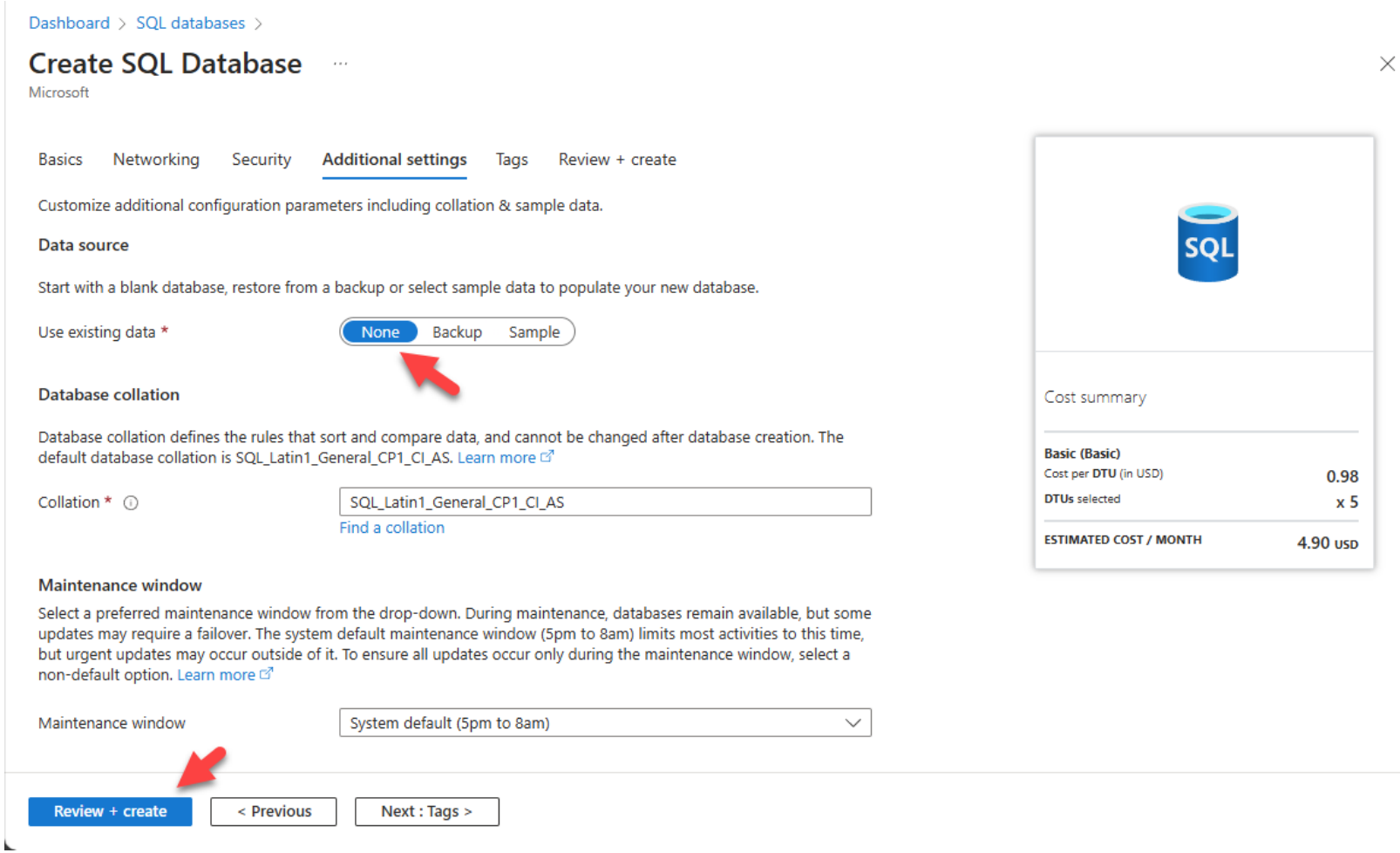
Azure SQL Basics



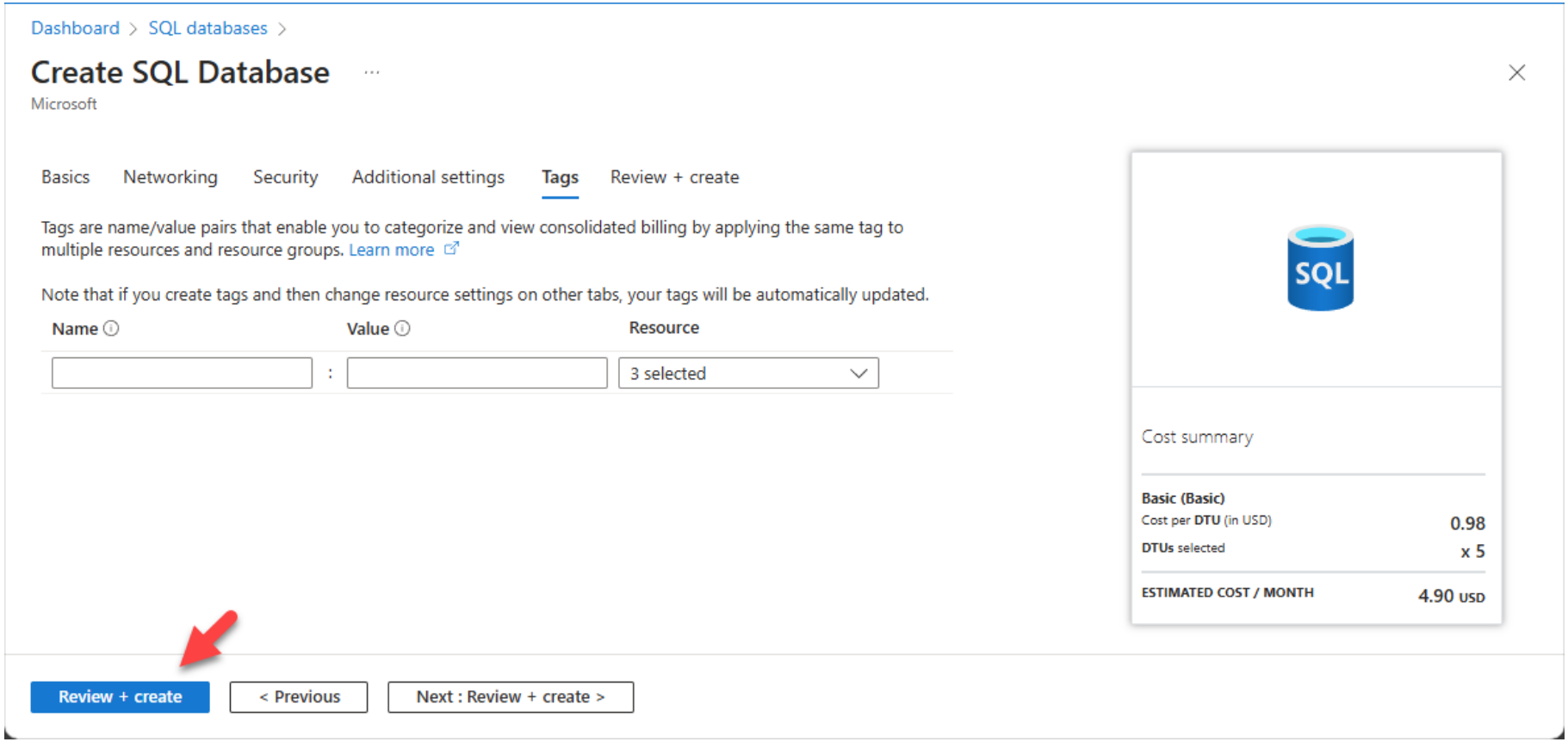
Azure SQL Basics



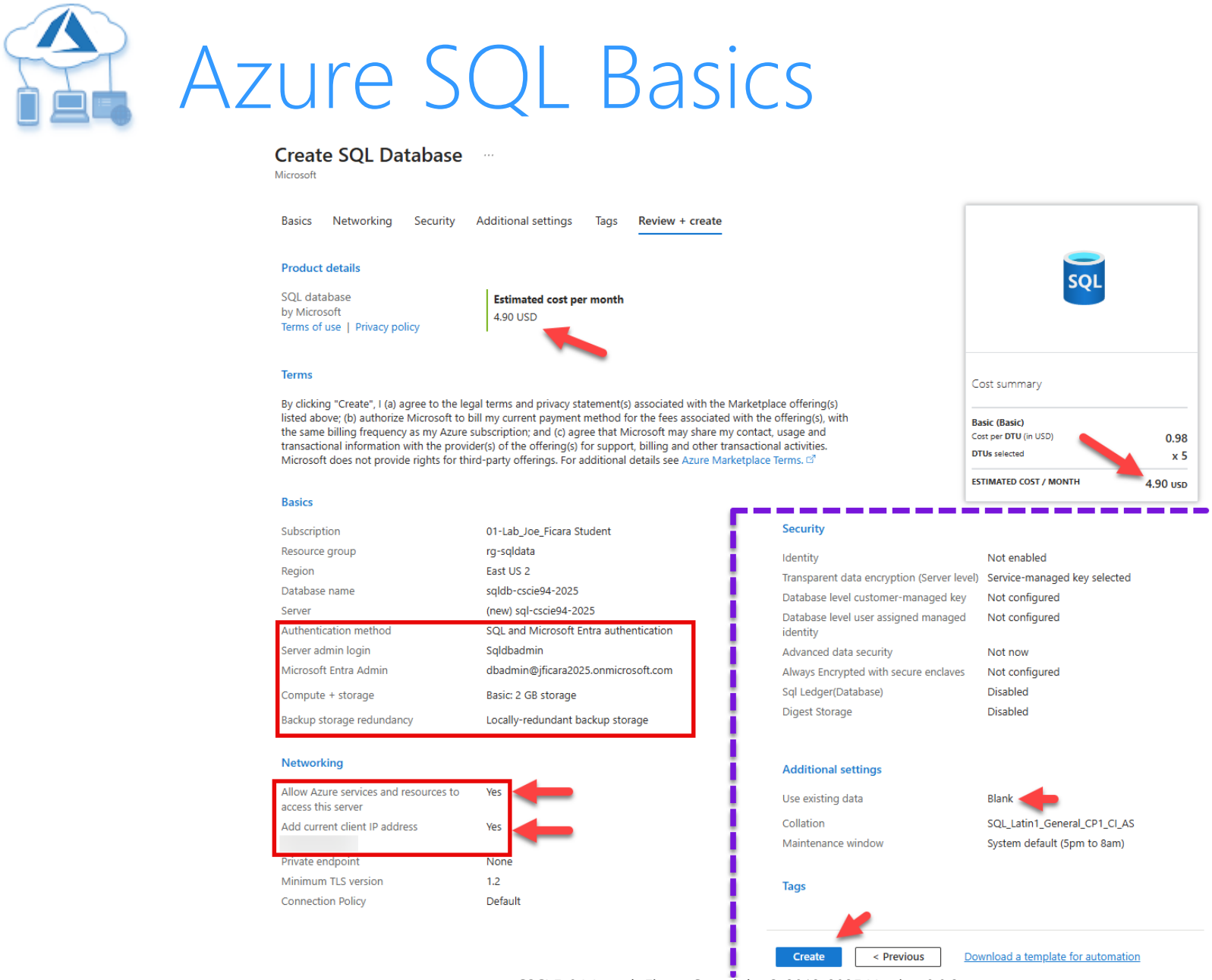
Azure SQL Basics



Azure SQL Basics



Azure SQL Basics

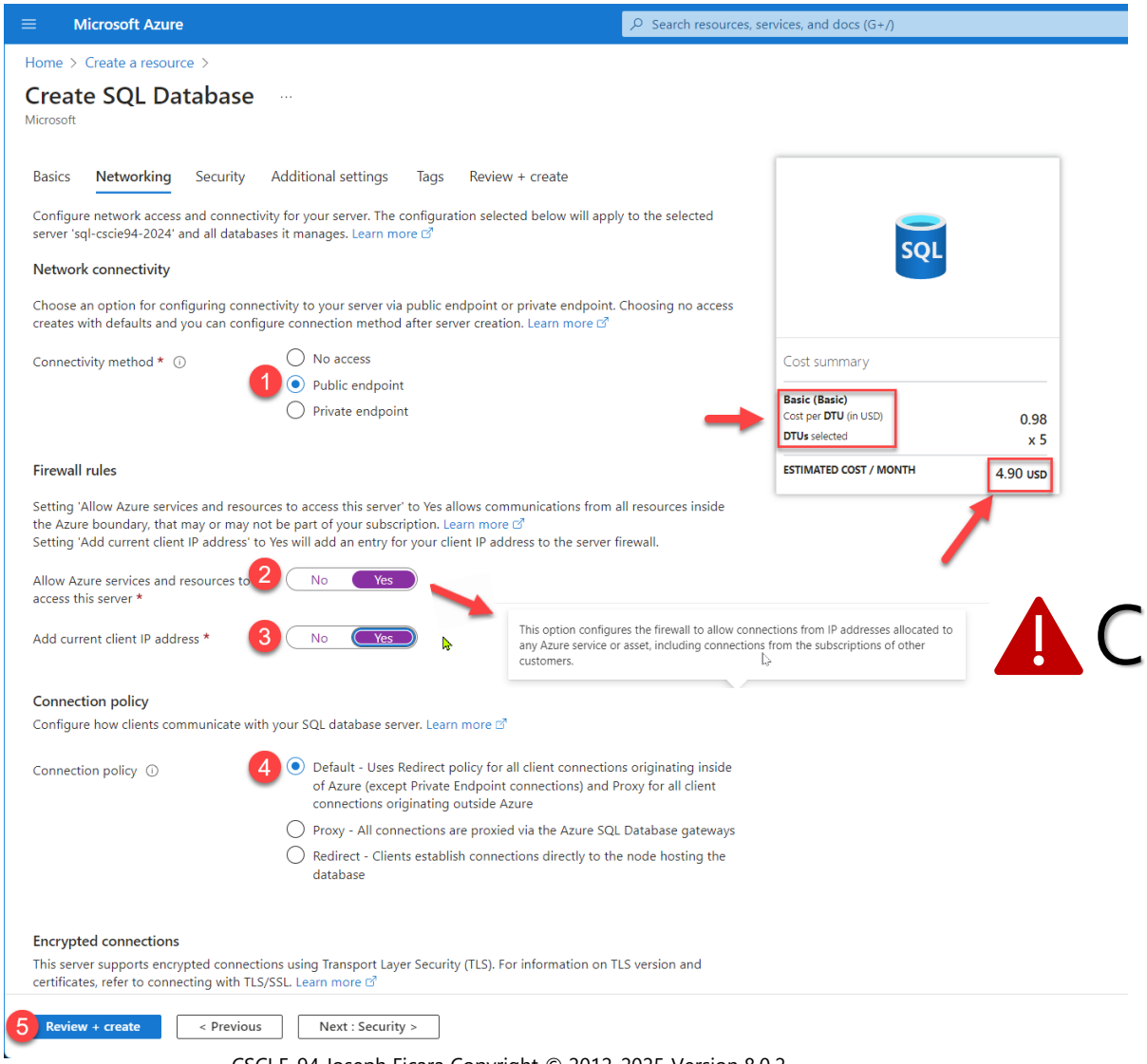


Azure SQL Basics

- If you forgot during…
  - Modify your firewall rules
    - On the Azure SQL Server
  - Allow access to
    - Your client IP Address
    - Azure services and resources
      - Any service in Azure can "*reach*" your SQL Server
        - From other subscriptions of other customers
        - Services **must have valid have credentials**
          - Defense in depth reduced with this option turned on
        - Alternative -> Network segmentation

Caution

Azure SQL Basics

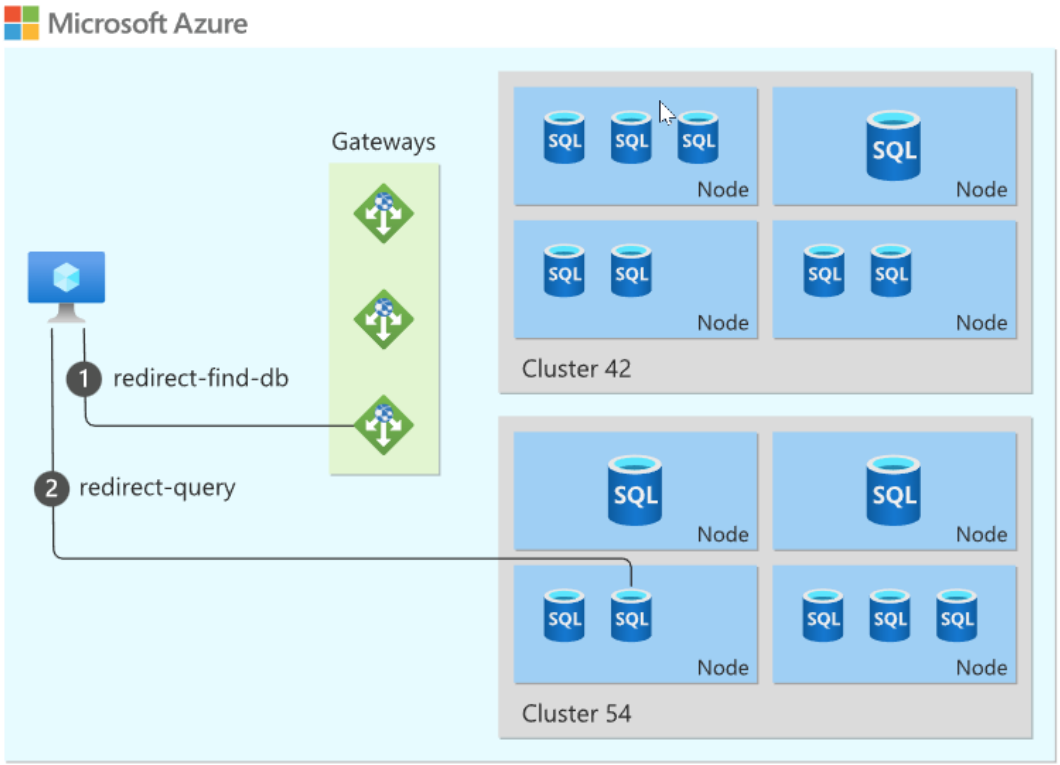


Caution

Publishing to Azure

Redirect Policy (Default)

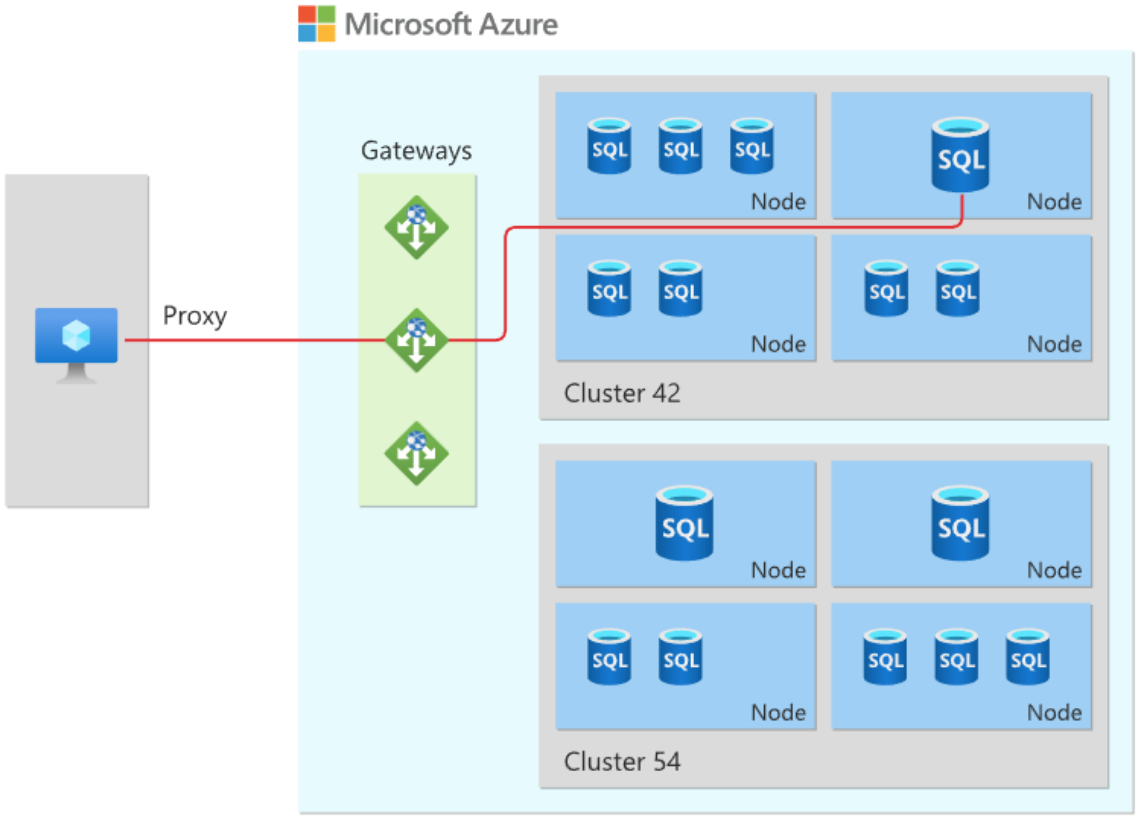
- Connectivity from within Azure



Publishing to Azure

Redirect Policy (Default)

- Connectivity from outside azure



Azure SQL Basics

- You can access the database via
  - Visual Studio
    - Server Explorer
      - Add database connection
  - Azure Data Studio
    - Linux, Mac, Windows
  - SQL Server Management Studio
    - Download here: http://bit.ly/2GPIpFF

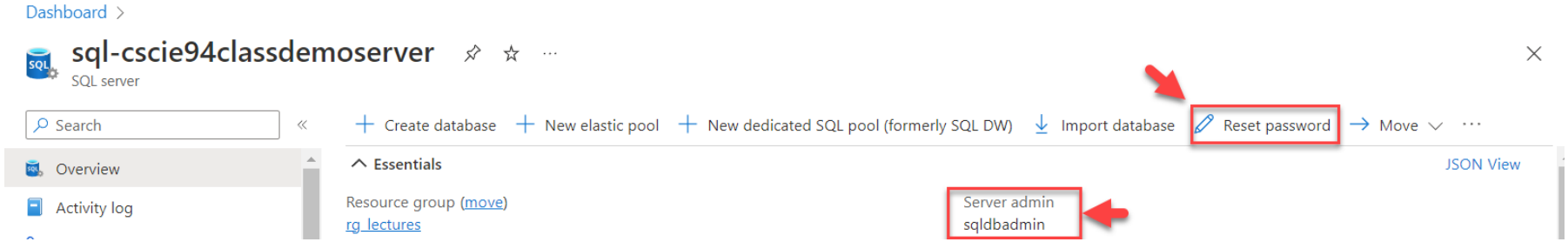
Demo

Azure SQL Basics

- Azure Portal
- SQL Management Studio
- Azure Data Studio

Azure SQL Basics

- How do I
  - Reset my Azure SQL Server password?
  - Find my Azure SQL Server Admin Id



Azure SQL Basics

Entity Framework Core Basics

- What is Entity Framework Core?
  - It's an Object Relational Mapping technology
  - Many similarities to Entity Framework 6
    - But built from scratch for .NET Core
      - See: http://bit.ly/2S8zxTg
    - For differences between EF 6 and EF Core
      - See: http://bit.ly/2Fq2KQp
  - Conceptually like other ORMs like
    - Hibernate
    - nHibernate

Entity Framework Core Basics

- Provider support for popular databases
  - SQL Server
  - My SQL
  - PostgreSQL
  - Oracle - For purchase
  - SQLite
  - SQL Compact - Only works on .NET Framework
  - DB2
  - In-memory (for testing)

Entity Framework Core Basics

- Model generation support
  - Reverse engineer existing database
    - First time reverse engineer only
    - Update model still not there ☹
      - Update model from database · Issue #831 · dotnet/efcore (github.com)
  - Generate model from C# classes
    - Code First

Entity Framework Core Basics

- Note **E**ntity **F**ramework 6 Fans…
  - In EF Core there is no:
    - EDMX file
      - Not planned
    - Relationship visualization
      - "High priority feature" in backlog
      - EF Core Power Tools is an option
        - See: Home · ErikEJ/EFCorePowerTools Wiki (github.com)
- EF Core roadmap
  - See: http://bit.ly/2RVvRB4

Entity Framework Core Basics

- Can you use EF Core with .NET Core?
  - Yes, with caveat
- How?
  - Create .NET Core projects
    - Using .NET 4.8 framework as option
  - .NET 4.8 is .NET Standard 2.0 compatible
    - .NET Standard 2.0 defines an API surface area
      - It's a specification that represents:
        - A set of APIs that all .NET platforms have to implement.
    - See .NET Standard FAQ: https://bit.ly/3uwtZ8x

Entity Framework Core Basics

- Let's build a .NET 9 Web API app that Uses
  - .NET 9 Web API
  - Entity Framework Core 9
  - Azure SQL
  - Creates, Updates, Retrieves and Deletes
    - Customer entities
    - Address entities
      - Customer has a 1 to many relationships to Address

Entity Framework Core Basics

- Basic support requires 8 steps
  - 1. Create a .NET 9 Web API project
  - 2. Add the Nuget Package
    - Microsoft.EntityFrameworkCore.SqlServer (9.0.x)
  - 3. Create a Customer entity
    - Annotate it with what property represents the key
  - 4. Create a Database in Azure SQL
  - 5. Add connection string information
    - To Web API App (Both (a) local and (b) Azure)
      - Note don't keep Azure permanently just for testing!

Connection Strings

Local connection string

```
// Local: Step 5a
"ConnectionStrings": {
  "DefaultConnection": "Server=(localdb)\\mssqllocaldb;
                       Database=<databasename>;
                       Trusted_Connection=True;
                       MultipleActiveResultSets=true",   }

// Example:
"ConnectionStrings": {
  "DefaultConnection": "Server=(localdb)\\mssqllocaldb;
                        Database=cscie94-efdemo-2025;
                        Trusted_Connection=True;
                        MultipleActiveResultSets=true" }
```

Connection Strings

Azure SQL Authentication connection string

```
DON'T KEEP THESE SETTINGS IN YOUR CONFIG FILE
PUT THEM IN THE AZURE PORTAL UNDER CONNECTION STRINGS

// Azure: Step 5b
// Obtained from the portal
"ConnectionStrings": {
  "DefaultConnection": "Server=tcp:<servername>,1433;
        Initial Catalog=<databaseName>;
        Persist Security Info=False;
        User ID={sql user id};
        Password={sql password};
        MultipleActiveResultSets=False;
        Encrypt=True;
        TrustServerCertificate=False;
        Connection Timeout=30;"   },
```

Connection Strings

Azure Microsoft Entra ID Authentication connection string

```
Active Directory Authentication
Uses identity associated with the "process"
"Connectionstrings": {
  "Defaultconnection": "Server= tcp:<servername>,1433;
                        Initial Catalog=<your database  name>;
                        Encrypt=True;
                        TrustServerCertificate=False;
                        Connection Timeout=30;
                        Authentication=Active Directory Default" }
Example
"Connectionstrings": {
  "Defaultconnection": "Server=tcp:sql-cscie94-2025.database.windows.net,1433;
                        Initial Catalog=sqldb-cscie94-efdemo-2025;
                        Encrypt=True;
                        TrustServerCertificate=False;
                        Connection Timeout=30;
                        Authentication=Active Directory Default" }
```

Entity Framework Core Basics

Basic support requires 8 steps…

- 6. Create a class that derives from DbContext
  - a) Creates the database if it does not exist
  - b) Add a DbSet<Customer> Customers property
    - Used to indicate creation of Customer table
  - C) Override OnModelCreating
    - To configure table to entity linking
- 7. Create a DBInitializer class
  - To seed the database with data
- 8. Update the program.cs
  - To setup the service provider with your context
    - This is the class created in step 6

Entity Framework Core Basics

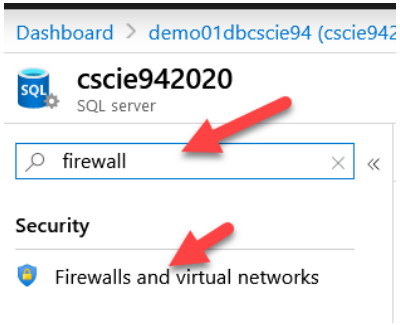
- What do we have now?
  - Project that will
    - Create a database
    - Create a Customer table
    - Populate Customer table with seed data
  - Work with Azure SQL or LocalDb
- What's next?
  - Wire up controller methods
    - To Create, Update, Get and Delete Data

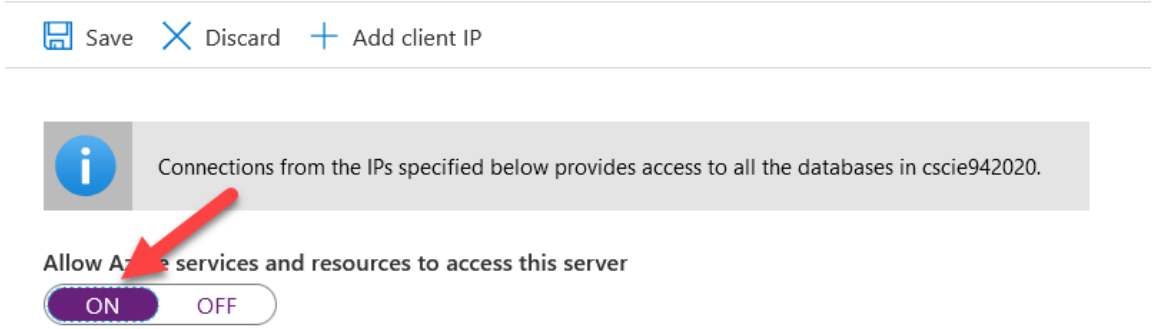
Entity Framework Core Basics

- In your Controller
  - 1. Modify Constructor
    - To take your class that's derived from DbContext
    - Save the value in a local private variable
  - 2. Create payload and result classes
    - Are used for input (payload) and result definitions
    - These enforce validation using attributes
    - Expose only public properties
  - Use LINQ and the context instance
    - To Create, Update, Delete, and Retrieve data

Publishing to Azure

- Ensure your database server allows access
  - By default, its set not to
    - Go to the database server
    - Type "firewall"
    - Click on the "Firewalls and virtual networks"
    - Set:
      - Allow Azure services and resources to access this server
      - ON





Publishing to Azure

⚠ Caution

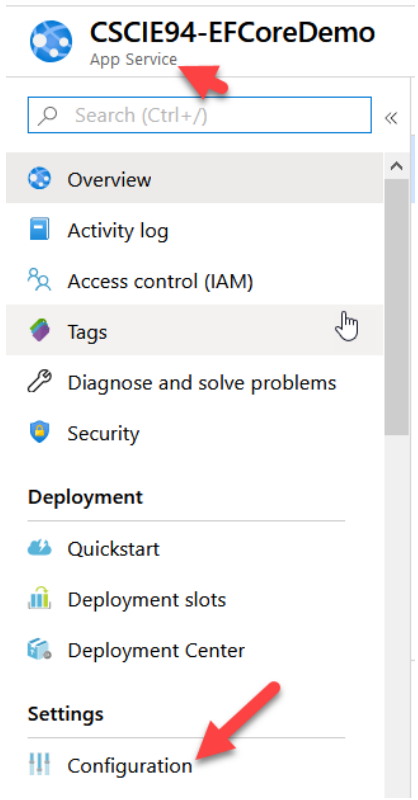
- Any service in Azure can "*reach*" your SQL Server
  - From other subscriptions of other customers
  - Services must have valid have credentials
  - Defense in depth
    - Reduced with this option turned on
  - Alternative
    - Network segmentation

Publishing to Azure

- If using settings files…
  - Have an appsettings.**Production**.json file
    - Correctly configured as part of your project
  - Set the **application setting** in the Azure portal
    - **ASPNETCORE_ENVIRONMENT**
      - To **Production**
  - Set the **connection string** in the Azure portal
    - **DefaultConnection**
      - To the connection string from your database
      - Be sure to include the **id** and **password**

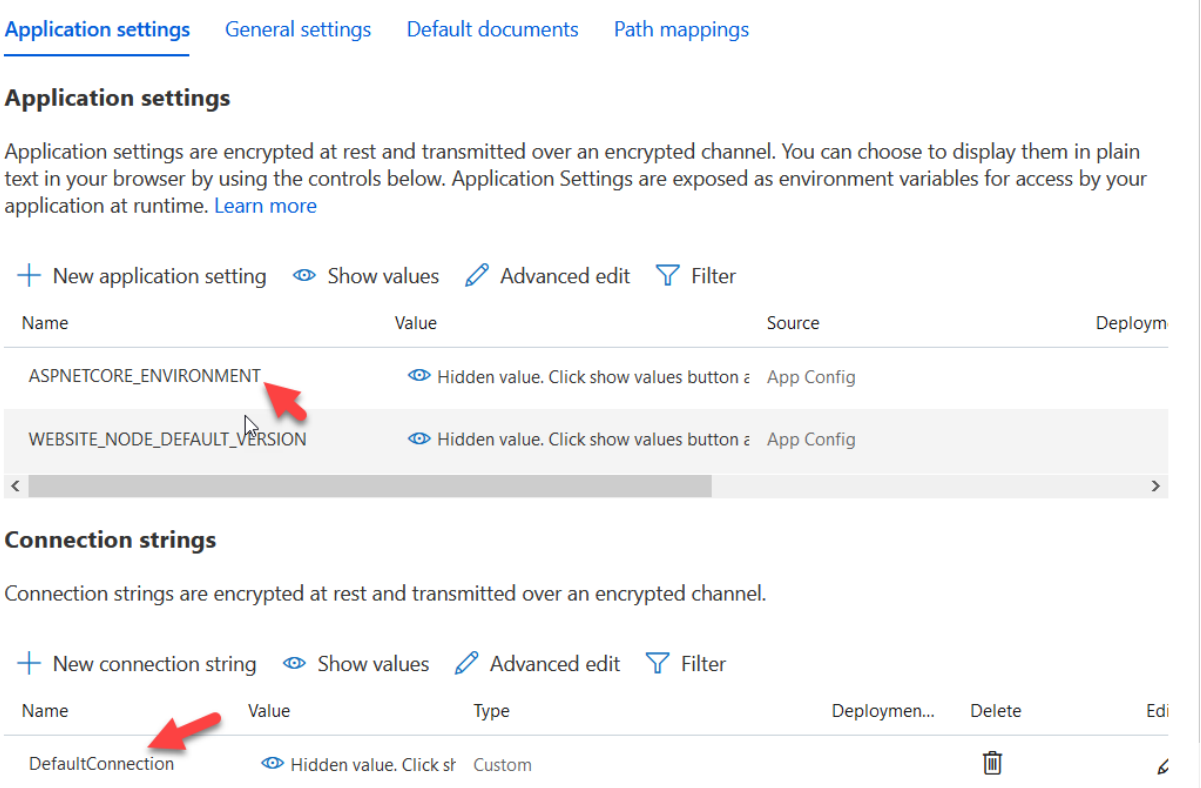
Publishing to Azure

- Better Option - Configuration Settings
  - Application settings & Connection string



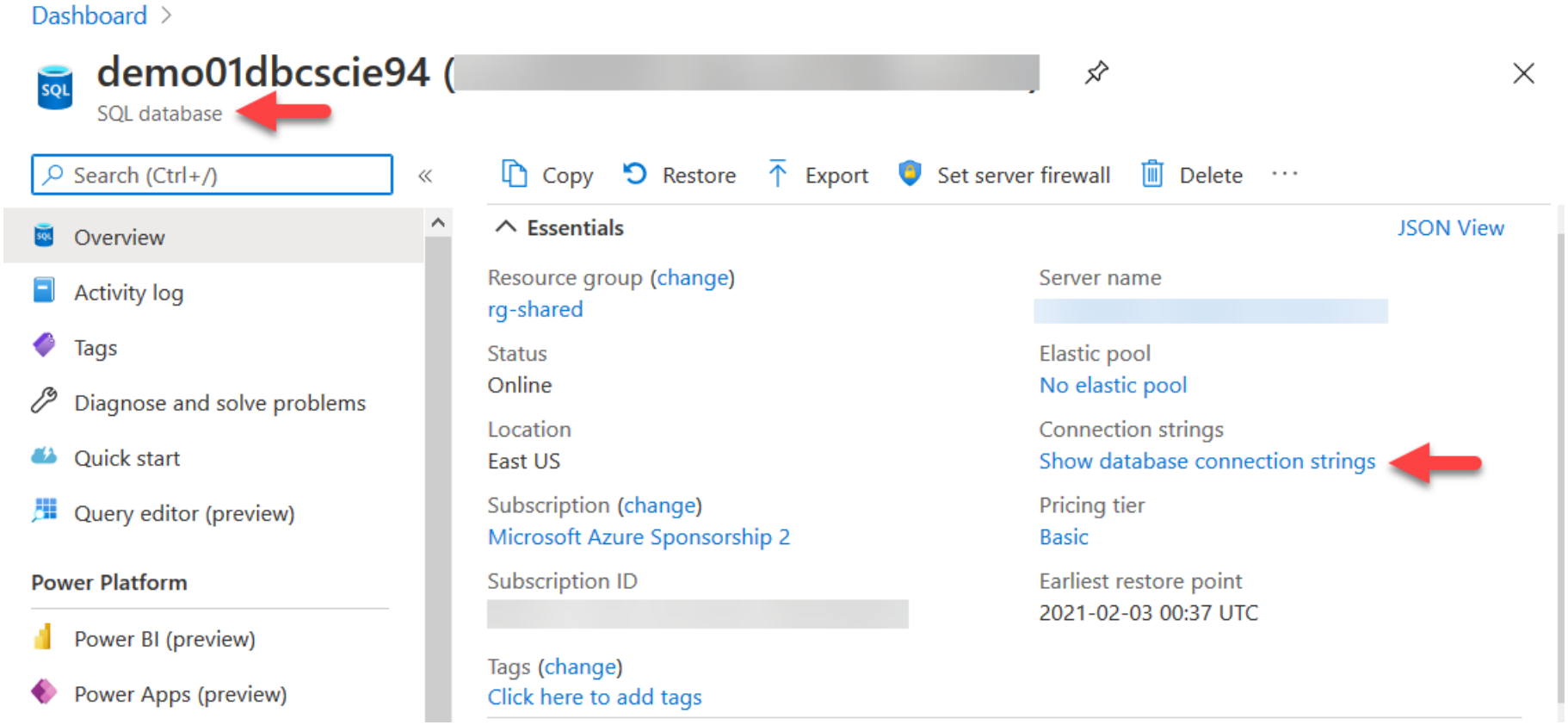
Publishing to Azure

- Better Option - Configuration Settings
  - Application settings & Connection string



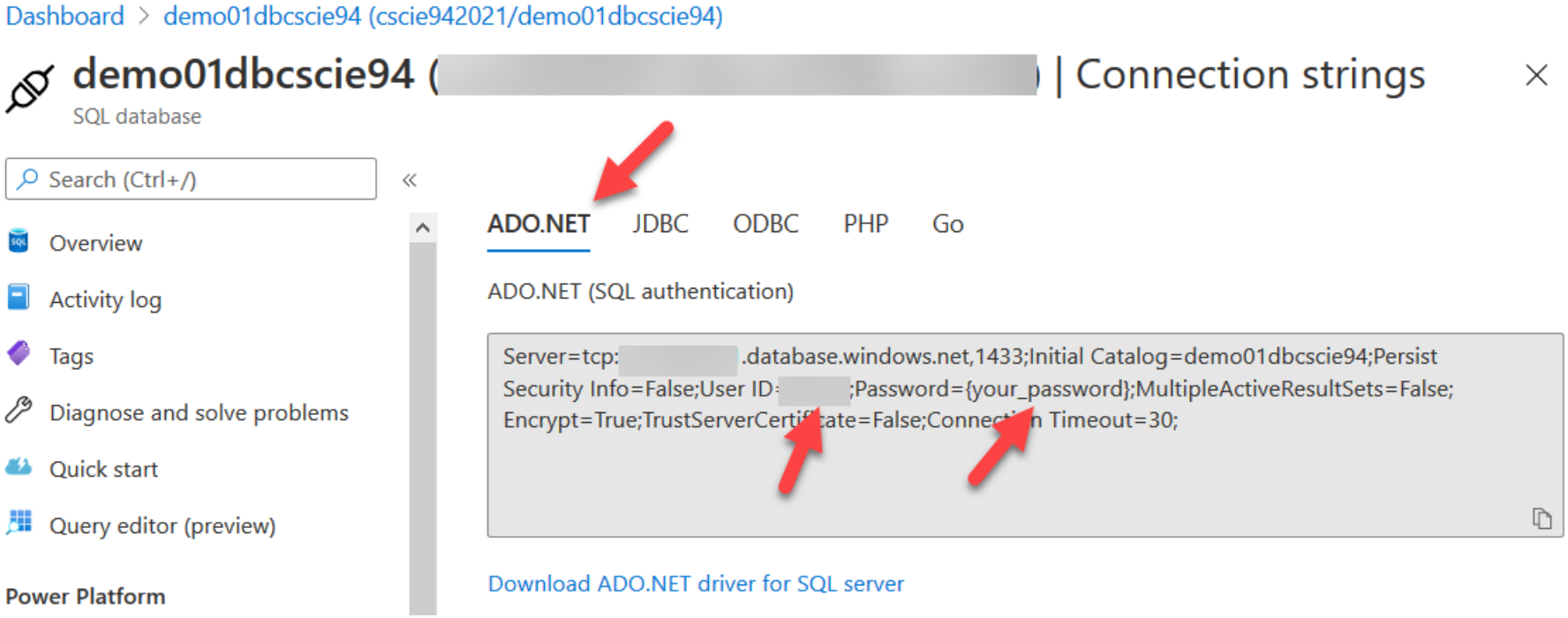
Publishing to Azure

- Database connection string is found here



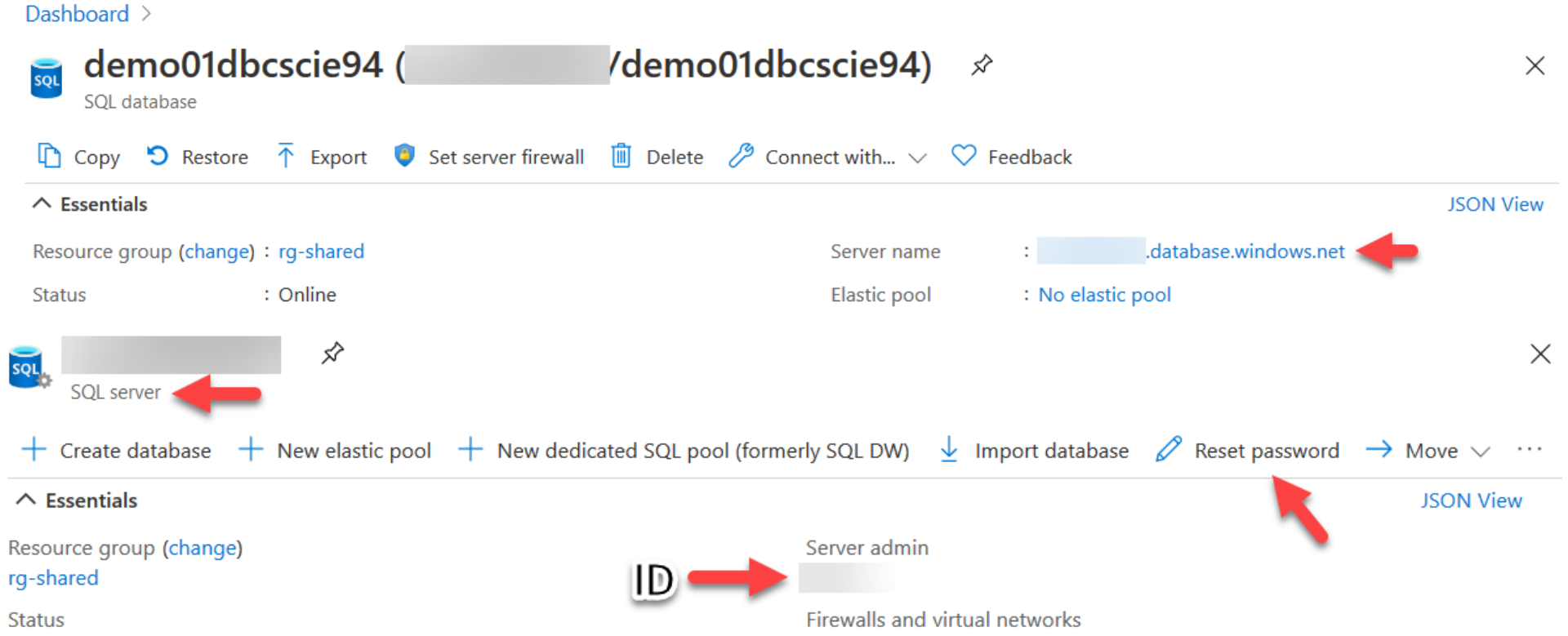
Publishing to Azure

- Database connection string is found here



Publishing to Azure

- ID can be found on the SQL Server blade
  - You can also reset the password here



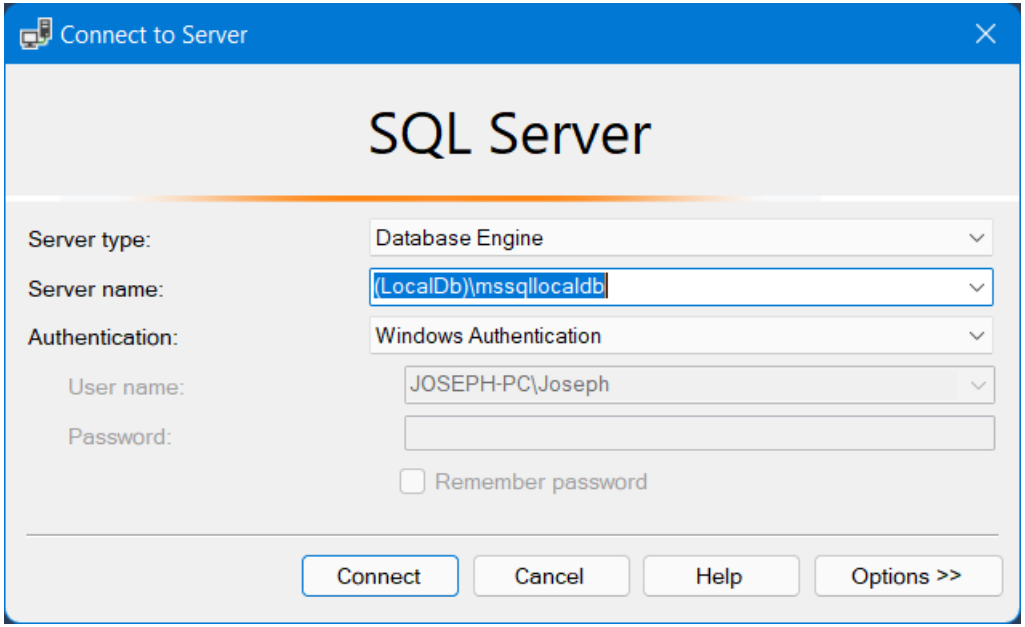
Demo

EntityFramework Demo

`EFCoreDemoSolution.sln`

SQL Server Management Studio
Tips

- Connecting to Local Db
  - (LocalDb)\mssqllocaldb



SQL Server Management Studio
Tips

- Excellent tool for
  - Ad-hoc queries
  - Modifying data
- To keep consistent
  - Use EF Core Code First to generate tables
- Quickest way to iterate
  - Drop table
    - Ensure you stop debugging
    - Choose close all connections

Demo

SQL Server Management Studio

Query Local and Azure DB

Azure Regions & Latency

- Latency
  - Data transfer time between regions
    - Time it takes to travel over the distance
- Charges for region-to-region data transfer
  - **Egress** is typically charged
    - Outbound traffic,
      - ~$0.02 USD per GB between regions in North America
        - See: Pricing - Bandwidth | Microsoft Azure
  - **Ingress** is not typically charged
    - Inbound traffic

Azure Regions & Latency

Courtesy Microsoft Documentation: [Azure network round-trip]

Source Region	Australia Central	Australia Central 2	Australia East	Australia Southeast	Brazil South	Canada Central	Canada East	Central India	Central US	East Asia	East US	East US 2	France Central	France South	Germany North	Germany West Central	Israel Central	Italy North	Japan East	Japan West	Korea Central	Korea South	North Central US	North Europe	Norway East	Norway West	Poland North	Qatar Central	South Africa North	South Africa West	South Central US	South India	Southeast Asia	Sweden Central	Switzerland North	Switzerland West	UAE Central	UAE North	UK South	UK West	West Central US	West Europe	West India	West US	West US 2	West US 3
Australia Central		3	10	18	304	201	211	148	177	127	203	198	285	274	296	292	313	288	108	114	132	129	190	296	308	304	305	164	280	295	166	131	101	312	284	284	178	182	287	290	164	291	173	148	168	150
Australia Central 2	4		9	15	304	200	211	149	177	127	203	198	278	267	288	284	307	282	107	114	132	129	190	286	300	294	297	183	281	294	166	131	100	305	278	279	178	181	276	278	164	281	172	148	167	149
Australia East	10	9		16	316	196	206	147	174	123	201	197	293	267	297	296	306	281	103	110	131	135	184	291	302	296	291	162	289	294	162	121	91	296	278	279	170	173	180	288	284	160	287	170	140	161	146
Australia Southeast	19	15	16		312	207	216	137	185	119	213	207	266	255	277	272	294	267	115	122	143	137	19																			171	273	160	153	174	157
Brazil South	305	304	317	312		131	136	307	149	323	119	118	190	186	204	199	226	196	278	280	303	309	13																			163	186	322	186	183	165
Canada Central	201	199	194	205	130		15	213	24	205	20	23	102	98	115	110	136	110	158	165	235	177	1																			38	95	235	61	60	66
Canada East	210	209	204	214	135	15		218	33	215	29	32	111	107	124	119	142	117	167	174	191	187	2																			48	104	239	71	69	75
Central India	148	147	147	135	307	213	218		231	87	203	205	126	131	140	135	158	137	121	128	118	111	22																			24	135	29	220	212	232
Central US	178	177	174	185	149	25	34	233		184	33	37	116	116	130	126	155	128	137	144	198	160	1																			18	116	254	41	39	46
East Asia	124	123	120	114	320	203	212	84	180		213	206	212	201	223	219	241	216	49	42	40	33	19																			167	219	1	151	143	155
East US	200	199	201	211	116	18	28	200	31	215		9	67	63	101	96	123	96	169	175	224	185	21	87	105	97	111	194	225	208	35	224	233	117	94	90	182	184	77	82	45	89	222	71	66	59
East US 2	196	195	198	204	116	23	33	206	35	208	10		84	88	97	92	128	100	160	167	188	180	26	96	108	101	108	198	230	213	34	228	232	116	99	95	186	189	88	91	50	91	225	69	70	53
France Central	287	278	294	266	190	103	113	127	116	217	90	85		14	18	13	54	25	251	257	247	240	107	20	31	25	27	125	157	141	112	148	186	35	16	13	114	116	10	13	131	12	147	150	150	137
France South	275	267	268	255	186	99	109	132	116	206	85	89	15		26	21	44	16	244	249	236	229	102	30	41	35	35	114	155	138	116	154	181	43	15	10	102	105	20	23	130	22	153	153	148	140
Germany North	297	288	298	278	204	117	126	142	130	228	103	99	19	26		13	67	28	264	267	258	252	121	35	23	27	13	138	171	154	128	159	195	21	16	18	125	128	25	27	145	16	162	165	164	154
Germany West Central	290	282	296	270	197	110	119	134	123	220	96	82	11	19	11		59	20	250	261	251	244	114	27	25	25	20	130	164	148	122	151	188	28	8	11	118	120	17	19	138	11	153	160	157	146
Israel Central	314	307	307	295	226	138	144	159	156	245	126	130	55	44	67	61		51	278	284	276	269	144	72	83	81	69	155	196	183	155	174	213	85	48	51	142	145	61	64	171	69	182	194	190	173
Italy North	286	278	279	266	197	110	117	138	127	218	97	101	24	15	25	20	50		249	256	248	241	115	42	38	37	34	126	168	150	126	159	188	41	13	13	113	116	31	35	141	25	154	165	162	147
Japan East	108	107	104	115	278	159	169	122	137	52	170	163	252	244	263	252	277	252		11	32	26	148	253	272	264	276	156	253	267	128	104	73	279	254	250	152	155	244	245	123	251	143	108	100	111
Japan West	115	114	110	122	281	166	176	129	144	46	177	170	257	249	267	263	284	259	12		19	13	155	259	276	268	279	163	261	274	134	111	73	283	262	258	160	162	249	252	130	254	150	115	107	118
Korea Central	133	132	134	144	302	236	193	119	197	43	229	198	247	236	258	254	276	250	33	19		9	218	263	271	267	267	153	251	265	166	101	70	275	247	249	149	152	252	255	179	254	140	160	124	148
Korea South	129	129	136	137	308	179	188	113	160	37	188	182	241	229	252	247	269	243	27	13	10		174	256	265	260	261	146	244	258	152	95	64	268	240	236	144	146	246	249	149	248	134	132	124	133
North Central US	190	190	185	195	137	18	28	225	14	195	24	27	107	102	121	115	144	117	149	155	218	173		105	127	117	131	214	246	228	37	248	213	132	114	110	202	205	94	98	28	109	243	51	49	55
North Europe	296	283	290	281	187	99	106	146	112	229	88	95	19	28	33	28	70	41	252	257	261	254	104		39	34	39	139	172	155	122	160	198	43	33	27	127	129	13	16	132	19	164	156	152	138
Norway East	309	300	303	290	211	114	124	154	136	240	108	110	32	41	23	27	83	39	272	276	271	265	127	40		10	31	150	184	167	136	170	208	11	30	33	138	140	30	34	150	25	176	172	171	158
Norway West	305	294	307	286	199	110	115	148	129	236	102	103	25	35	27	27	81	39	264	269	267	260	118	35	11		35	146	177	160	133	162	204	17	30	33	134	136	25	27	145	18	172	167	167	149
Poland North	306	297	303	286	210	123	129	151	141	237	113	112	28	35	13	22	69	34	277	279	267	261	129	41	29	34		146	180	163	141	165	205	26	25	27	134	136	31	35	155	23	173	175	177	159
Qatar Central	184	182	182	170	296	208	213	41	226	121	196	199	124	114	136	131	153	126	155	161	152	145	214	140	149	144	145		118	131	224	55	90	153	125	122	16	17	130	133	240	132	65	264	254	243
South Africa North	280	278	279	266	326	239	244	138	257	218	227	230	156	154	170	164	195	167	252	259	250	241	245	172	183	176	179	117		20	255	153	188	166	165	161	105	102	162	165	271	163	158	294	291	274
South Africa West	295	294	295	282	311	223	228	153	241	233	210	214	141	138	154	151	183	152	268	274	265	257	228	158	167	160	163	132	21		239	168	203	170	148	145	121	118	147	151	255	147	174	278	275	258
South Central US	166	165	162	173	142	48	58	230	27	174	38	33	110	114	127	123	153	126	127	134	165	150	36	123	136	132	137	224	256	238		230	204	141	127	124	212	214	114	117	26	118	252	36	48	22
South India	132	131	131	119	326	237	242	26	232	70	225	230	148	155	159	153	173	162	104	111	101	94	247	162	170	162	165	55	154	169	230		38	173	163	150	52	56	149	156	223	156	44	203	196	215
Southeast Asia	100	98	95	86	343	223	233	55	200	38	234	233	183	179	193	189	211	190	72	75	69	62	212	198	206	202	202	88	187	201	203	36		210	185	184	85	88	188	190	190	90	77	171	163	187
Sweden Central	313	305	307	294	213	125	131	158	142	244	119	117	35	43	21	30	84	42	279	283	275	268	132	44	11	17	27	154	188	170	142	173	212		32	35	141	144	34	39	162	27	180	183	182	166
Switzerland North	287	280	281	268	199	111	119	141	129	219	99	103	18	17	18	12	50	15	256	263	249	242	117	36	31	32	27	128	168	150	127	164	189	35		8	115	118	25	29	143	19	161	167	164	148
Switzerland West	285	279	280	262	193	106	116	133	123	217	93	97	13	10	19	14	51	15	251	258	249	236	111	28	33	33	28	124	163	145	125	151	187	36	7		110	112	18	21	138	20	154	160	159	148
UAE Central	178	177	178	166	284	196	202	35	214	119	184	188	115	103	125	120	142	115	152	160	149	144	202	130	137	133	134	17	108	121	213	53	87	141	113	110		6	119	122	229	122	56	252	249	231
UAE North	183	180	180	168	287	199	204	38	217	121	187	190	116	104	127	122	144	117	155	162	152	145	205	131	140	135	136	18	104	118	215	56	90	144	116	112	6		122	124	231	123	62	255	251	234
UK South	287	275	281	272	179	90	99	133	104	221	79	89	10	20	24	19	61	32	244	249	252	245	94	14	30	24	31	131	163	146	114	149	189	33	24	18	119	121		8	124	11	155	148	146	133
UK West	289	277	284	274	181	93	102	138	105	223	83	92	13	23	26	20	64	35	245	251	255	248	97	17	33	25	34	133	165	150	117	155	191	38	28	20	120	123	7		127	15	158	153	149	136
West Central US	164	163	160	171	162	39	49	240	17	170	47	51	129	129	142	140	170	141	123	129	154	149	27	132	149	142	154	240	271	254	25	222	188	161	141	136	227	230	123	127		132	262	27	25	34
West Europe	291	279	286	272	185	95	102	136	119	222	90	91	11	21	16	12	67	24	250	253	253	246	109	20	23	17	24	132	163	146	119	160	190	27	16	19	121	122	10	13	132		155	154	154	136
West US	148	147	140	153	186	62	71	221	41	154	74	67	149	153	163	162	193	165	108	114	159	131	50	159	170	166	173	263	295	277	36	203	172	182	163	159	251	254	147	152	27	155	242		24	19
West US 2	169	168	162	174	184	62	71	214	39	147	69	73	151	149	164	160	190	163	100	107	124	124	49	155	170	167	176	255	292	275	50	196	165	181	162	158	249	251	147	150	26	154	234	25		40
West US 3	149	148	145	156	163	66	75	232	45	158	60	56	136	139	152	147	171	147	111	117	147	132	53	143	157	148	159	242	274	256	22	214	188	165	150	146	229	232	133	136	34	139	253	19	39	

East US to West US
71 ms as of June 20, 204 2024

Azure Regions & Latency

- Some resources
  - Must exist in the same region
    - For performance
    - Compliance
    - Operation

Azure Regions & Latency

- Some examples:
  - App service & App Service Plan
  - Azure Functions & App Service Plan
  - Azure Logic Apps and Logic App Workflows
  - Resources within the same virtual network
    - Different VNETs can be connected across regions

Azure Regions & Latency

- Recommendation
  - Keep **high traffic** resources in same region
    - App Service and Database its connected to
  - Application Insights
    - Same region as the services sending data to it
      - Reduces cost and latency

Azure Regions & Latency

- Can resource groups contain resources in different regions?
  - Yes
- Why would you do this?
  - Management
  - Billing
  - Access control
  - Unified lifecycle
  - Cross-region applications

Azure Regions & Latency

- Reminder… Latency
  - Data transfer time between regions
    - Time it takes to travel over the distance
- Let's look at a demo
  - Remember this number …
    - **71ms** West Coast to East Coast

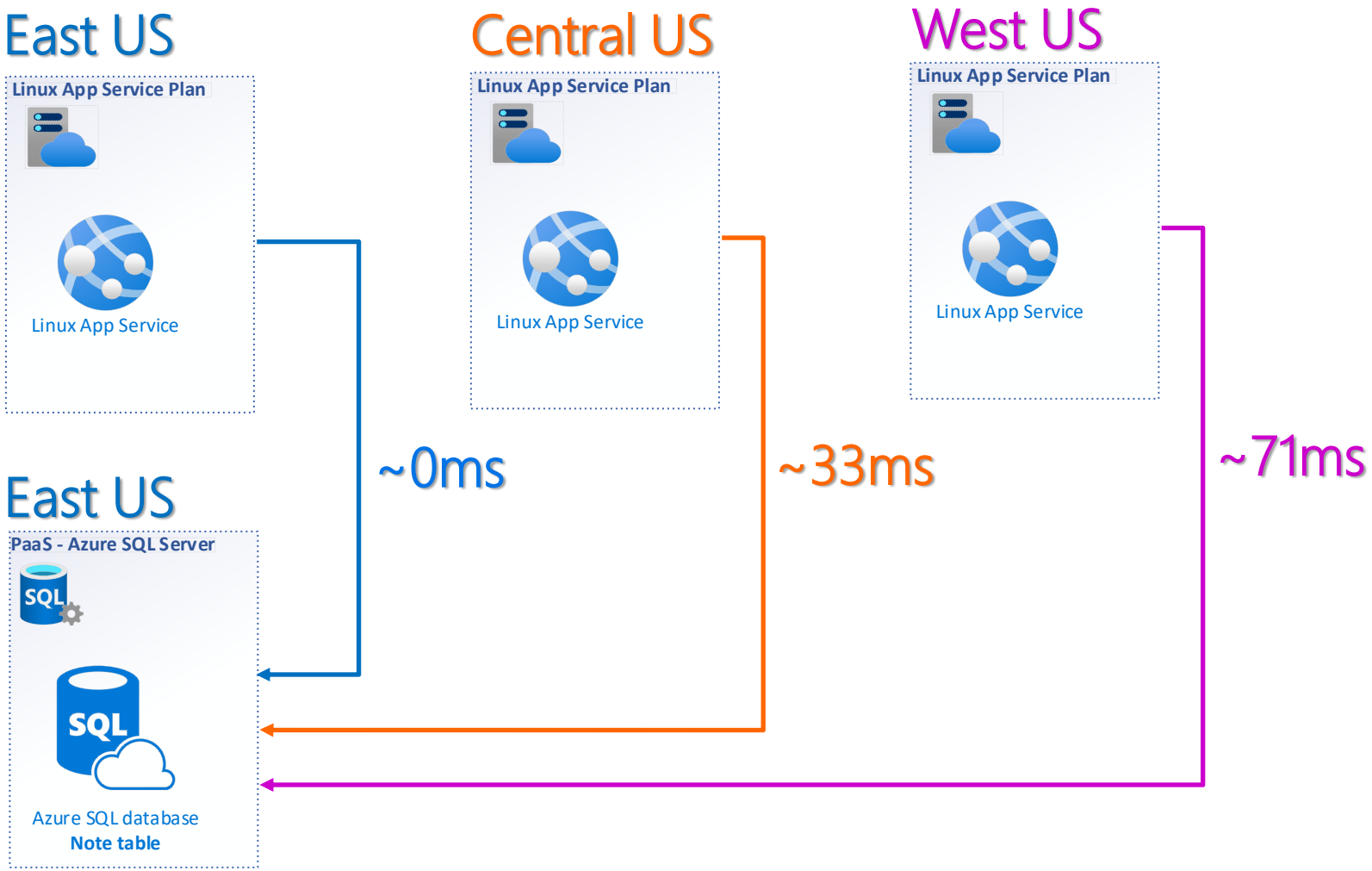
Azure Regions & Latency

Courtesy Microsoft Documentation: [Azure network round-trip]

Source Region	Australia Central	Australia Central 2	Australia East	Australia Southeast	Brazil South	Canada Central	Canada East	Central India	Central US	East Asia	East US	East US 2	France Central	France South	Germany North	Germany West Central	Israel Central	Italy North	Japan East	Japan West	Korea Central	Korea South	North Central US	North Europe	Norway East	Norway West	Poland North	Qatar Central	South Africa North	South Africa West	South Central US	South India	Southeast Asia	Sweden Central	Switzerland North	Switzerland West	UAE Central	UAE North	UK South	UK West	West Central US	West Europe	West India	West US	West US 2	West US 3
Australia Central		3	10	18	304	201	211	148	177	127	203	198	285	274	296	292	313	288	108	114	132	129	190	296	308	304	305	184	280	295	166	131	101	312	284	284	178	182	287	290	164	291	173	148	168	150
Australia Central 2	4		9	15	304	200	211	149	177	127	203	198	278	267	288	284	307	282	107	114	132	129	190	286	300	294	297	183	281	294	166	131	100	305	278	279	178	181	276	278	164	281	172	148	167	149
Australia East	10	9		16	316	196	206	147	174	123	201	197	293	267	297	296	306	281	103	110	131	135	184																			160	287	170	140	161	146
Australia Southeast	19	15	16		312	207	216	137	185	119	213	207	266	255	277	272	294	267	115	122	143	137	19																			171	273	160	153	174	157
Brazil South	305	304	317	312		131	136	307	149	323	119	118	190	186	204	199	226	198	278	280	303	309	13																			163	186	322	186	183	165
Canada Central	201	199	194	205	130		15	213	24	205	20	23	102	98	115	110	136	110	158	165	235	177	1																			38	95	235	61	60	66
Canada East	210	209	204	214	135	15		218	33	215	29	32	111	107	124	119	142	117	167	174	191	187	2																			48	104	239	71	69	75
Central India	148	147	147	135	307	213	218		231	87	203	205	126	131	140	135	158	137	121	128	118	111	22																			248	135	29	220	212	232
Central US	178	177	174	185	149	25	34	233		184	33	37	116	116	130	126	155	128	137	144	198	160	1																			18	132	254	41	39	46
East Asia	124	123	120	114	320	203	212	84	180		213	206	212	201	223	219	241	216	49	42	40	33	19																			167	219	1	151	143	155
East US	200	199	201	211	116	18	28	200	31	215		9	87	83	101	96	123	96	169	175	224	185	21	87	105	97	111	194	225	208	35	224	233	117	94	90	182	184	77	82	45	89	222	71	66	59
East US 2	196	195	198	204	116	23	33	206	35	208	10		84	88	97	92	128	100	160	167	188	180	26	96	108	101	108	198	230	213	34	228	232	116	99	95	186	189	88	91	50	91	225	69	70	53
France Central	287	278	294	266	190	103	113	127	116	217	90	85		14	18	13	54	25	251	257	247	240	107	20	31	25	27	125	157	141	112	148	186	35	16	13	114	116	10	13	131	12	147	150	150	137
France South	275	267	268	255	186	99	109	132	116	206	85	89	15		26	21	44	16	244	249	236	229	102	30	41	35	35	114	155	138	116	154	181	43	15	10	102	105	20	23	130	22	153	153	148	140
Germany North	297	288	298	278	204	117	126	142	130	228	103	99	19	26		13	67	28	264	267	258	252	121	35	23	27	13	138	171	154	128	159	195	21	16	18	125	128	25	27	145	16	162	165	164	154
Germany West Central	290	282	296	270	197	110	119	134	123	220	96	92	11	19	11		59	20	250	261	251	244	114	27	25	25	20	130	164	148	122	151	188	28	8	11	118	120	17	19	138	11	153	160	157	146
Israel Central	314	307	307	295	226	138	144	159	156	245	126	130	55	44	67	61		51	278	284	276	269	144	72	83	81	69	155	196	183	155	174	213	85	48	51	142	145	61	64	171	69	182	194	190	173
Italy North	286	278	279	266	197	110	117	138	127	218	97	101	24	15	25	20	50		249	256	248	241	115	42	38	37	34	126	168	150	126	159	188	41	13	13	113	116	31	35	141	25	154	165	162	147
Japan East	108	107	104	115	278	159	169	122	137	52	170	163	252	244	263	252	277	252		11	32	26	148	253	272	264	276	156	253	267	128	104	73	279	254	250	152	155	244	245	123	251	143	108	100	111
Japan West	115	114	110	122	281	166	176	129	144	46	177	170	257	249	267	263	284	259	12		19	13	155	259	276	268	279	163	261	274	134	111	73	283	262	258	160	162	249	252	130	254	150	115	107	118
Korea Central	133	132	134	144	302	236	193	119	197	43	229	198	247	236	258	254	278	250	33	19		9	218	263	271	267	267	153	251	265	166	101	70	275	247	249	149	152	252	255	179	254	140	160	124	148
Korea South	129	129	136	137	308	179	188	113	160	37	188	182	241	229	252	247	269	243	27	13	10		174	256	265	260	261	146	244	258	152	95	64	268	240	236	144	146	246	249	149	248	134	132	124	133
North Central US	190	190	185	195	137	18	28	225	14	195	24	27	107	102	121	115	144	117	149	155	218	173		105	127	117	131	214	246	228	37	248	213	132	114	110	202	205	94	98	28	109	243	51	49	55
North Europe	296	283	290	281	187	99	106	146	112	229	88	95	19	28	33	28	70	41	252	257	261	254	104		39	34	39	139	172	155	122	160	198	43	33	27	127	129	13	16	132	19	164	156	152	138
Norway East	309	300	303	290	211	114	124	154	136	240	108	110	32	41	23	27	83	39	272	276	271	265	127	40		10	31	150	184	167	136	170	208	11	30	33	138	140	30	34	150	25	176	172	171	158
Norway West	305	294	307	286	199	110	115	148	129	236	102	103	25	35	27	27	81	39	264	269	267	260	118	35	11		35	146	177	160	133	162	204	17	30	33	134	136	25	27	145	18	172	167	167	149
Poland North	306	297	303	286	210	123	129	151	141	237	113	112	28	35	13	22	69	34	277	279	267	261	129	41	29	34		146	180	163	141	165	205	26	25	27	134	136	31	35	155	23	173	175	177	159
Qatar Central	184	182	182	170	296	208	213	41	226	121	196	199	124	114	136	131	153	126	155	161	152	145	214	140	149	144	145		118	131	224	55	90	153	125	122	16	17	130	133	240	132	65	264	254	243
South Africa North	280	278	279	266	326	239	244	138	257	218	227	230	156	154	170	164	195	167	252	259	250	241	245	172	183	176	179	117		20	255	153	188	166	165	161	105	102	162	165	271	163	158	294	291	274
South Africa West	295	294	295	282	311	223	228	153	241	233	210	214	141	138	154	151	183	152	268	274	265	257	228	158	167	160	163	132	21		239	168	203	170	148	145	121	118	147	151	255	147	174	278	275	258
South Central US	166	165	162	173	142	48	58	230	27	174	38	33	110	114	127	123	153	126	127	134	165	150	36	123	136	132	137	224	256	238		230	204	141	127	124	212	214	114	117	26	118	252	36	48	22
South India	132	131	131	119	326	237	242	26	232	70	225	230	148	155	159	153	173	162	104	111	101	94	247	162	170	162	165	55	154	169	230		38	173	163	150	52	56	149	156	223	156	44	203	196	215
Southeast Asia	100	98	95	86	343	223	233	55	200	38	234	233	183	179	193	189	211	190	72	75	69	62	212	198	206	202	202	88	187	201	203	36		210	185	184	85	88	188	190	190	190	77	171	163	187
Sweden Central	313	305	307	294	213	125	131	158	142	244	119	117	35	43	21	30	84	42	279	283	275	268	132	44	11	17	27	154	188	170	142	173	212		32	35	141	144	34	39	162	27	160	183	182	166
Switzerland North	287	280	281	268	199	111	119	141	129	219	99	103	18	17	18	12	50	15	256	263	249	242	117	36	31	32	27	128	168	150	127	164	189	35		8	115	118	25	29	143	19	161	167	164	148
Switzerland West	285	279	280	262	193	106	116	133	123	217	93	97	13	10	19	14	51	15	251	258	249	236	111	28	33	33	28	124	163	145	125	151	187	36	7		110	112	18	21	138	20	154	160	159	148
UAE Central	178	177	178	166	284	196	202	35	214	119	184	188	115	103	125	120	142	115	152	160	149	144	202	130	137	133	134	17	108	121	213	53	87	141	113	110		6	119	122	229	122	56	252	249	231
UAE North	183	180	180	168	287	199	204	38	217	121	187	190	116	104	127	122	144	117	155	162	152	145	205	131	140	135	136	18	104	118	215	56	90	144	116	112	6		122	124	231	123	62	255	251	234
UK South	287	275	281	272	179	90	99	133	104	221	79	89	10	20	24	19	61	32	244	249	252	245	94	14	30	24	31	131	163	146	114	149	189	33	24	18	119	121		8	124	11	155	148	146	133
UK West	289	277	284	274	181	93	102	138	105	223	83	92	13	23	26	20	64	35	245	251	255	248	97	17	33	25	34	133	165	150	117	155	191	38	28	20	120	123	7		127	15	158	153	149	136
West Central US	164	163	160	171	162	39	49	240	17	170	47	51	129	129	142	140	170	141	123	129	154	149	27	132	149	142	154	240	271	254	25	222	188	161	141	136	227	230	123	127		132	262	27	25	34
West Europe	291	279	286	272	185	95	102	136	119	222	90	91	11	21	16	12	67	24	250	253	253	246	109	20	23	17	24	132	163	146	119	160	190	27	16	19	121	122	10	13	132		155	154	154	136
West US	148	147	140	153	186	62	71	221	41	154	74	67	149	153	163	162	193	165	108	114	159	131	50	159	170	166	173	263	295	277	36	203	172	182	163	159	251	254	147	152	27	155	242		24	19
West US 2	169	168	162	174	184	62	71	214	39	147	69	73	151	149	164	160	190	163	100	107	124	124	49	155	170	167	176	255	292	275	50	196	165	181	162	158	249	251	147	150	26	154	234	25		40
West US 3	149	148	145	156	163	66	75	232	45	158	60	56	136	139	152	147	171	147	111	117	147	132	53	143	157	148	159	242	274	256	22	214	188	165	150	146	229	232	133	136	34	139	253	19	39	

East US to West US
71 ms as of June 20, 204 2024

Azure Regions & Latency



Demo

Latency in action - Azure

`app-efcoredemo-eastus-cscie94`

`app-efcoredemo-westus-cscie94`

Azure Regions & Latency?

Strongly Typed Settings

- .NET Core - Strongly Typed Settings
  - Allow for
    - Grouping of settings
      - All Search related settings
      - All Limit related settings
    - Are strongly typed
      - Reduces effort in converting from strings
  - Are easy to setup!

Strongly Typed Settings

- Here are 7 steps
  - 1. Define your settings
    - Add them to JSON config file
  - 2. Define C# class that models JSON settings
  - 3. Instantiate your C# Class
  - 4. Bind it to the settings
  - 5. Use Dependency Injection
    - To add a singleton of instance from step 3

Strongly Typed Settings

- Here are 7 steps
  - 6. Update controller to take in your class
  - 7. Save a local instance
    - You access your settings via the saved instance

Strongly Typed Settings

Configure an App Service app in the Azure portal

- Notes: http://bit.ly/38UjpdR
  - To set a setting in the Azure Portal
    - Setting name is combination of
      - Section name and property name
    - Windows
      - **<section name>:<property name>**
        - Ex: **CustomerLimits:MaxCustomers**
    - Linux
      <section name>__<property name>
      - Ex: **CustomerLimits__MaxCustomers**
        - The __ is a double under score

Demo

Strongly Typed Settings

`EFCoreDemoSolution.sln`

Strongly Typed Settings

Further Reading

- Fundamentals of Azure Second Edition
  - Authors: Michael Collier, Robin Shahan
  - ISBN: 978-1-5093-0296-3
  - Chapters 6 (Databases)
    - See: http://bit.ly/2GCpikP

Further Reading

- Pro ASP.NET Core 7
  - Author: Adam Freeman
  - ISBN: 978-1633437812
  - Chapters 7,17,18,32

Further Reading

- Pro C# 10 with .NET 6
  - Author: Andrew Troelsen, Philip Japikse
  - ISBN: 978-1484278680
  - Chapters 21-23 (Entity Framework Core)
  - Chapters 30-34 (ASP.NET Core)

Links

- ASP.NET Core Validation
  - http://bit.ly/2nm7xeq
- Entity Framework Core
  - http://bit.ly/2GmCkQK
- ASP.NET Core Fundamentals
  - http://bit.ly/2BdLSdf
- Azure SQL
  - http://bit.ly/2DJz4RZ

Links

- Azure Data Studio
  - http://bit.ly/3XcRi1R
- SQL Server Management Studio
  - http://bit.ly/2GPIpFF