Challenges with Aurora PSQL

Andraz Brodnik aka. brodul

- 15+ YOE 👴
- Worked in AWS ☁️ back and forth from 2010
- Multiple roles trough my career 🃏 ➡️


{aurora}

Situation 1/2

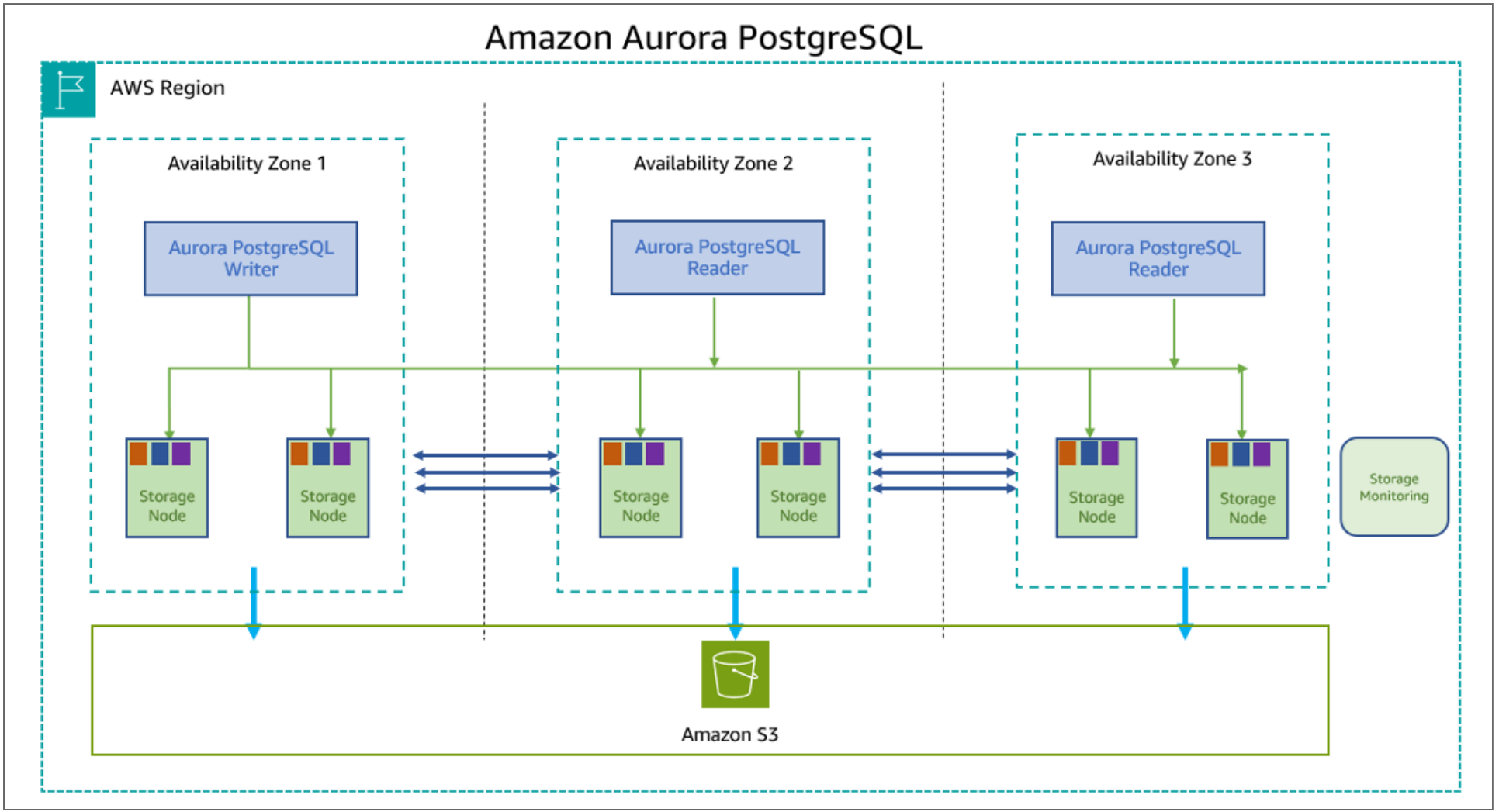
- Joined a company 👋
- Got tasked to move tenants from Cockroach DB to AWS RDS Aurora (Postgres engine) 🤓
- Each tenant own database 📥

Situation 2/2

- Cockroach is almost compatible Postgres database 👌
- Distributed across nodes (uses SSTables) 🪄
- StatefulSet in EKS 🪚
- Super amazing colleges (SRE and Platform) 👫

AWS RDS Aurora

- Database as a service 💸
- Proprietary / supports MySql and Postgres 🕵️
- 3x throughput 🐴
- Storage layer is abstracted away 💾 ✳️
- Synchronization over storage ☎️
- Cheaper 🤑


Amazon Aurora PostgreSQL
AWS Region
Availability Zone 1
Availability Zone 2
Availability Zone 3
Aurora PostgreSQL Writer
Aurora PostgreSQL Reader
Aurora PostgreSQL Reader
Storage Node
Storage Node
Storage Node
Storage Node
Storage Node
Storage Node
Storage Monitoring
Amazon S3

...

... it's one less thing we have to maintain 💯

Context

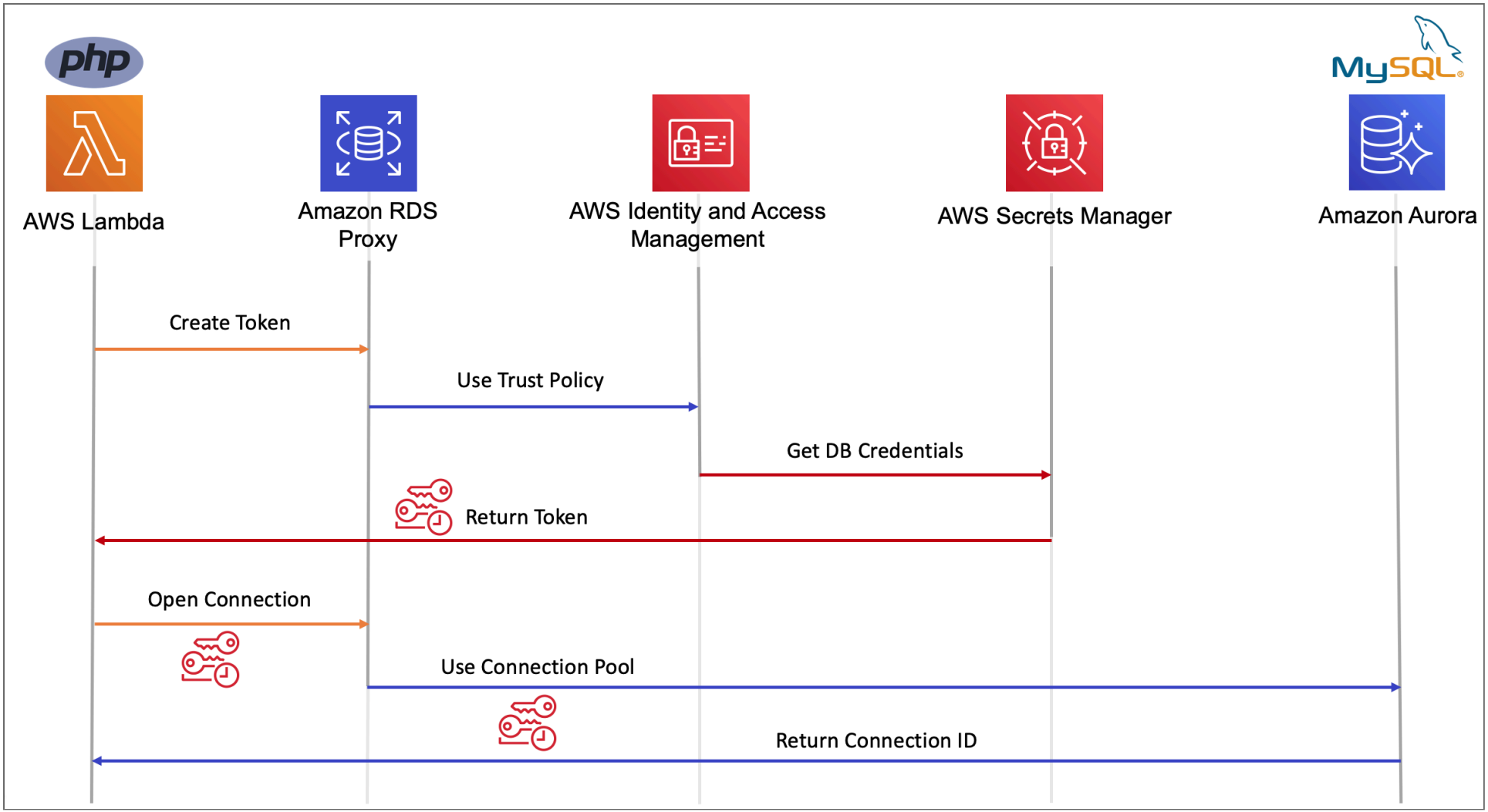
- use aurora-postgres engine
- use `provisioned` instead of `serverless`
- Terraform

Pricing 💸

- Storage is basically free
- You pay for the compute/instances and IOPS
- With IO optimized storage type you don't pay for IOPS, but you pay 30% premium on the instances

RDS proxy

- Connection pooler (PgBouncer) 👨‍👨‍👧
- Used to enable IAM auth, but that is now supported on basic Aurora


php
MySQL
AWS Lambda
Amazon RDS Proxy
AWS Identity and Access Management
AWS Secrets Manager
Amazon Aurora
Create Token
Use Trust Policy
Get DB Credentials
Return Token
Open Connection
Use Connection Pool
Return Connection ID

API call `modify rds proxy`

Auth parameter

```
[
  {
    "Description": "string",
    "UserName": "string",
    "AuthScheme": "SECRETS",
    "SecretArn": "string",
    "IAMAuth": "DISABLED"|"REQUIRED"|"ENABLED",
    "ClientPasswordAuthType":
        "POSTGRES_SCRAM_SHA_256"|"POSTGRES_MD5"
  }
  ...
]
```

RDS Proxy Limits

- 200 users max per proxy
- Proxy goes into modifying state

Naming / API namespacing

- Everything is RDS 🥇🥇
- RDS CreateDBCluster api call for example

CreateDBCluster

PDF

Creates a new Amazon Aurora DB cluster or Multi-AZ DB cluster.

If you create an Aurora DB cluster, the request creates an empty cluster. You must explicitly create the writer instance for your DB cluster using the CreateDBInstance operation. If you create a Multi-AZ DB cluster, the request creates a writer and two reader DB instances for you, each in a different Availability Zone.

You can use the `ReplicationSourceIdentifier` parameter to create an Amazon Aurora DB cluster as a read replica of another DB cluster or Amazon RDS for MySQL or PostgreSQL DB instance. For more information about Amazon Aurora, see What is Amazon Aurora? in the *Amazon Aurora User Guide*.

You can also use the `ReplicationSourceIdentifier` parameter to create a Multi-AZ DB cluster read replica with an RDS for MySQL or PostgreSQL DB instance as the source. For more information about Multi-AZ DB clusters, see Multi-AZ DB cluster deployments in the *Amazon RDS User Guide*.

```python
def create_cluster(is_aurora=False, **args):
    if is_aurora:
        something()
    elif not is_aurora:
        something_else()
```

Resource: aws_rds_cluster

Manages a RDS Aurora Cluster or a RDS Multi-AZ DB Cluster. To manage cluster instances that inherit configuration from the cluster (when not running the cluster in `serverless` engine mode), see the `aws_rds_cluster_instance` resource. To manage non-Aurora DB instances (e.g., MySQL, PostgreSQL, SQL Server, etc.), see the `aws_db_instance` resource.

For information on the difference between the available Aurora MySQL engines see Comparison between Aurora MySQL 1 and Aurora MySQL 2 in the Amazon RDS User Guide.

Changes to an RDS Cluster can occur when you manually change a parameter, such as `port`, and are reflected in the next maintenance window. Because of this, Terraform may report a difference in its planning phase because a modification has not yet taken place. You can use the `apply_immediately` flag to instruct the service to apply the change immediately (see documentation below).

> ⚠ **Note:**
>
> Multi-AZ DB clusters are supported only for the MySQL and PostgreSQL DB engines

Multi-AZ DB clusters are supported only for the MySQL and PostgreSQL DB engines.

> ⚠ **Note:**
>
> `ca_certificate_identifier` is only supported for Multi-AZ DB clusters.

> ⚠ **Note:**
>
> using `apply_immediately` can result in a brief downtime as the server reboots. See the AWS Docs on RDS Maintenance for more information.

> ⚠ **Note:**

This resource supports the following arguments:

- `allocated_storage` - (Optional, Required for Multi-AZ DB cluster) The amount of storage in gibibytes (GiB) to allocate to each DB instance in the Multi-AZ DB cluster.

- `storage_encrypted` - (Optional) Specifies whether the DB cluster is encrypted. The default is `false` for `provisioned` `engine_mode` and `true` for `serverless` `engine_mode`. When restoring an unencrypted `snapshot_identifier`, the `kms_key_id` argument must be provided to encrypt the restored cluster. Terraform will only perform drift detection if a configuration value is provided.
- `storage_type` - (Optional, Required for Multi-AZ DB cluster) (Forces new for Multi-AZ DB clusters) Specifies the storage type to be associated with the DB cluster. For Aurora DB clusters, `storage_type` modifications can be done in-place. For Multi-AZ DB Clusters, the `iops` argument must also be set. Valid values are: `""`, `aurora-iopt1` (Aurora DB Clusters); `io1`, `io2` (Multi-AZ DB Clusters). Default: `""` (Aurora DB Clusters); `io1` (Multi-AZ DB Clusters).

EnableCloudwatchLogsExports.member.N

The list of log types that need to be enabled for exporting to CloudWatch Logs.

Valid for Cluster Type: Aurora DB clusters and Multi-AZ DB clusters

The following values are valid for each DB engine:

- Aurora MySQL - `audit | error | general | instance | slowquery | iam-db-auth-error`
- Aurora PostgreSQL - `instance | postgresql | iam-db-auth-error`
- RDS for MySQL - `error | general | slowquery | iam-db-auth-error`
- RDS for PostgreSQL - `postgresql | upgrade | iam-db-auth-error`

For more information about exporting CloudWatch Logs for Amazon RDS, see Publishing Database Logs to Amazon CloudWatch Logs in the *Amazon RDS User Guide*.

For more information about exporting CloudWatch Logs for Amazon Aurora, see Publishing Database Logs to Amazon CloudWatch Logs in the *Amazon Aurora User Guide*.

Type: Array of strings

Required: No

ScalingConfiguration

For DB clusters in `serverless` DB engine mode, the scaling properties of the DB cluster.

Valid for Cluster Type: Aurora DB clusters only

Type: ScalingConfiguration object

Required: No

ScalingConfiguration

For DB clusters in `serverless` DB engine mode, the scaling properties of the DB cluster.

Valid for Cluster Type: Aurora DB clusters only

Type: ScalingConfiguration object

Required: No

ServerlessV2ScalingConfiguration

Contains the scaling configuration of an Aurora Serverless v2 DB cluster.

For more information, see Using Amazon Aurora Serverless v2 in the *Amazon Aurora User Guide*.

Type: ServerlessV2ScalingConfiguration object

Required: No

🥲

Everyone is a bit confused

Multi AZ #421

Closed

devasivaram opened on Dec 23, 2023

Hi, I have been using this module for a long time, but now I have a requirement to create RDS Aurora Postgresql with Multi-AZ. I have tried a lot to make modifications or use another module. If possible could you please create an Multi AZ RDS Aurora Postgresql terraform code?

bryantbiggs on Dec 23, 2023 Member

Aurora is multi-az by default
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.AuroraHighAvailability.html

bryantbiggs closed this as completed on Dec 23, 2023

devasivaram on Dec 23, 2023 Author

But I can choose the multi az enable option while creating it from the console, but via Terraform creation when I check the configuration it shows multi az as no. Even I have tried to add 2 instances as well the configuration status of multi az shows no while created by Terraform, but from console, it will show Yes.

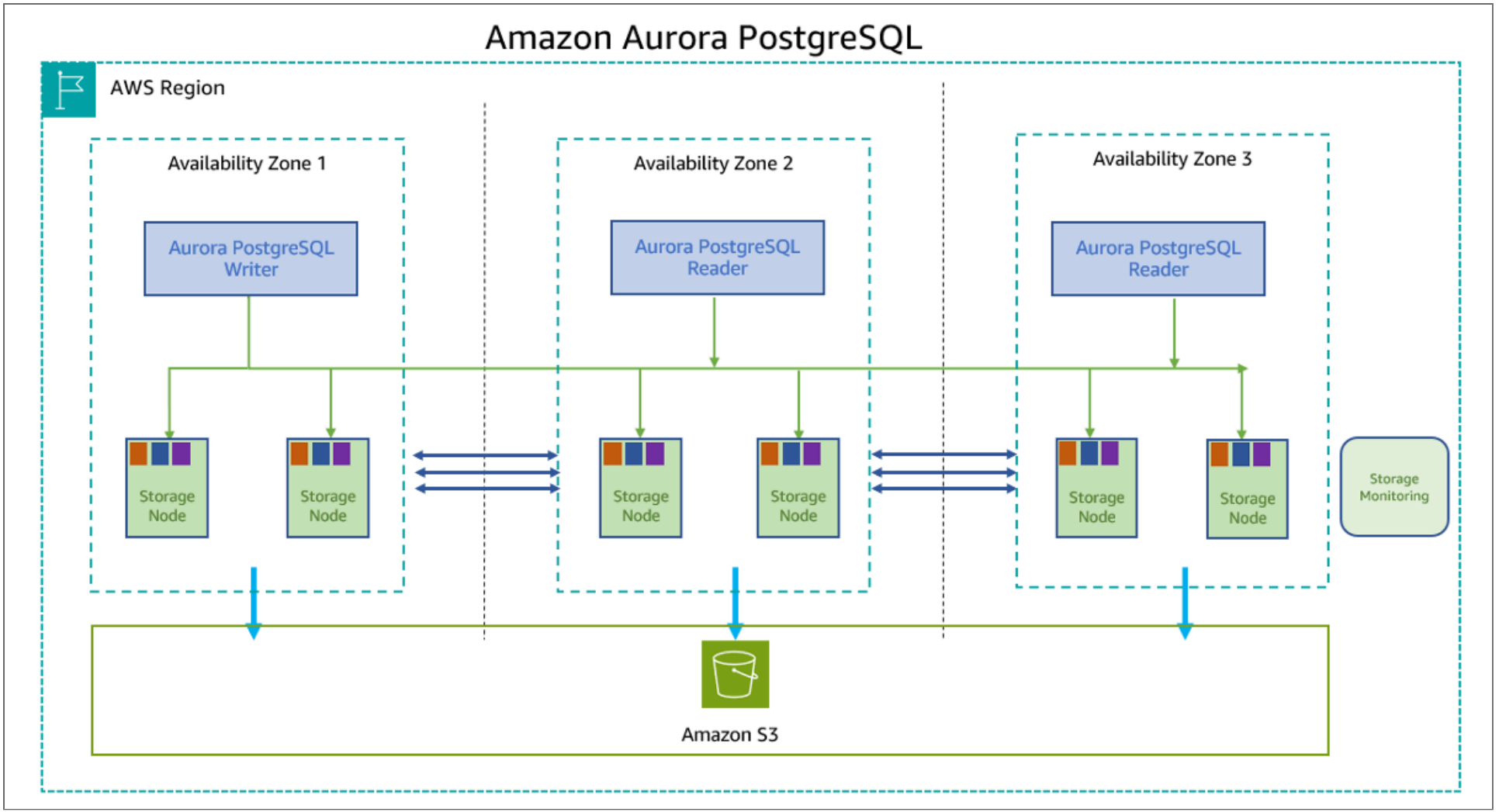
ahummel25 on Jan 4, 2024

@devasivaram **@bryantbiggs** It seems like the option to place additional instances of your Aurora cluster in separate AZs is not an option here. So Aurora itself might be Multi AZ compatible by default, but with this module, the cluster instances can only reside in a single AZ and cannot be scattered across AZs?

bryantbiggs on Jan 5, 2024 Member

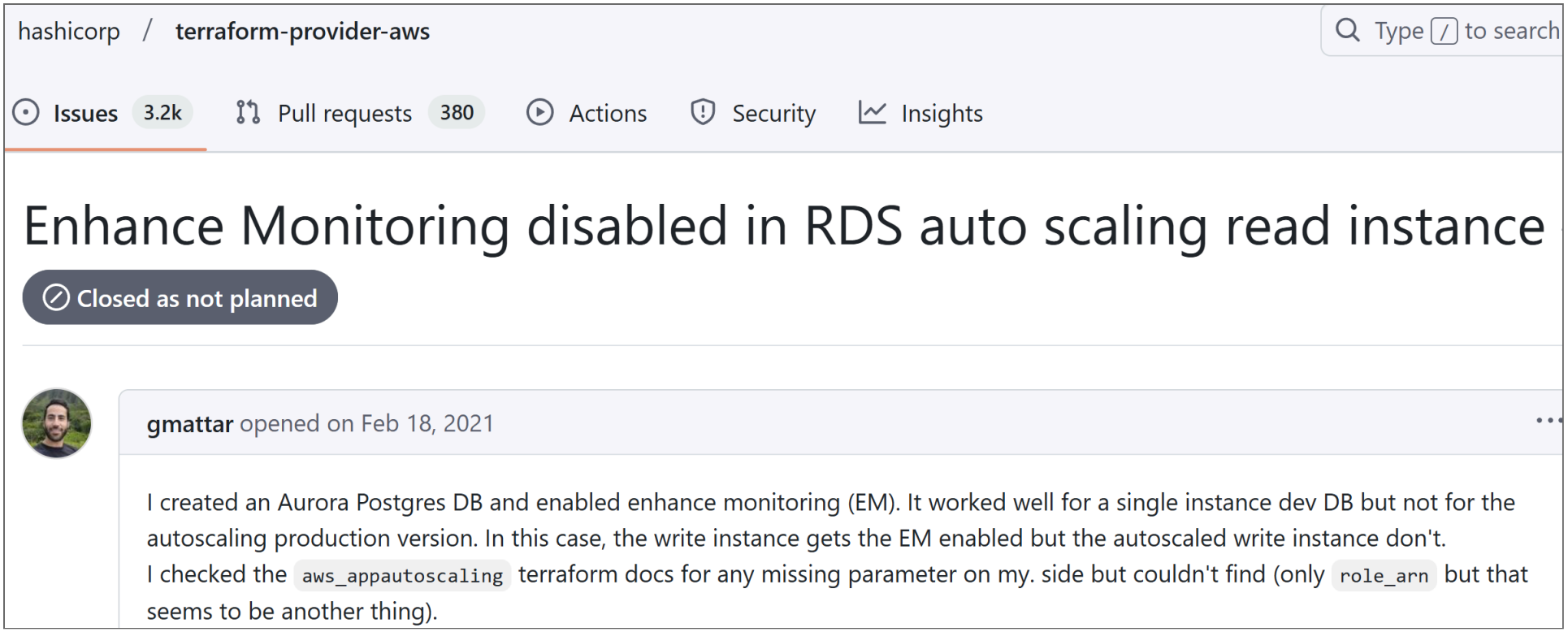
> the cluster instances can only reside in a single AZ and cannot be scattered across AZs

Thats not true


Amazon Aurora PostgreSQL
AWS Region
Availability Zone 1
Availability Zone 2
Availability Zone 3
Aurora PostgreSQL Writer
Aurora PostgreSQL Reader
Aurora PostgreSQL Reader
Storage Node
Storage Node
Storage Node
Storage Node
Storage Node
Storage Node
Storage Monitoring
Amazon S3

```
<DeletionProtection>false</DeletionProtection>
<Endpoint>my-multi-az-cluster.cluster-123456789012.us-west-2.rds.amazonaws.com</Endpoint>
<EngineMode>provisioned</EngineMode>
<Engine>mysql</Engine>
<ReaderEndpoint>my-multi-az-cluster.cluster-ro-123456789012.us-west-2.rds.amazonaws.com</ReaderEnd
<PubliclyAccessible>true</PubliclyAccessible>
<IAMDatabaseAuthenticationEnabled>false</IAMDatabaseAuthenticationEnabled>
<ClusterCreateTime>2021-10-20T00:12:00.867Z</ClusterCreateTime>
<PerformanceInsightsEnabled>true</PerformanceInsightsEnabled>
<MultiAZ>true</MultiAZ>
<DomainMemberships />
<MonitoringRoleArn>arn:aws:iam::123456789012:role/enhance-monitoring-role</MonitoringRoleArn>
<StorageEncrypted>true</StorageEncrypted>
<DBSubnetGroup>mysubnet1</DBSubnetGroup>
<VpcSecurityGroups>
    <VpcSecurityGroupMembership>
        <VpcSecurityGroupId>sg-6921cc28</VpcSecurityGroupId>
        <Status>active</Status>
    </VpcSecurityGroupMembership>
</VpcSecurityGroups>
<TagList />
<HostedZoneId>Z3GZ3VYA3PGHTO</HostedZoneId>
```

```
def create_cluster(is_aurora=False, **args):
    if is_aurora:
        something()
        if context_web_console:
            multi_az = True
```


hashicorp / terraform-provider-aws
Type / to search
Issues 3.2k
Pull requests 380
Actions
Security
Insights
Enhance Monitoring disabled in RDS auto scaling read instance
Closed as not planned
gmattar opened on Feb 18, 2021
I created an Aurora Postgres DB and enabled enhance monitoring (EM). It worked well for a single instance dev DB but not for the autoscaling production version. In this case, the write instance gets the EM enabled but the autoscaled write instance don't.
I checked the aws_appautoscaling terraform docs for any missing parameter on my. side but couldn't find (only role_arn but that seems to be another thing).

Postgres challenge

Postgres supports UTF8 encoding ...

Can you save any Unicode character as UTF8 encoding?

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

`NULL` char is used by Postgres internally

Thank you

Speaker notes