CONTINUITY SOFTWARE

aws partner network
Advanced Technology Partner
Marketplace seller

HealthCheck Sample Report

AvailabilityGuard NXG for Public Cloud (aka, Project Coral)

September 2019

By: Continuity Software

Revision: v2.2

For: [CUSTOMER]

Table of contents

Introduction

To address the resilience challenges of public cloud environments, **[CUSTOMER]** invited Continuity Software to check a subset of the production environment using our new SaaS solution, code-named Project Coral.

About Project Coral

You can prevent costly outages and assure resilience for your public cloud infrastructure and services (starting with AWS and Azure) with the help of Coral, the new SaaS solution from Continuity Software.

You have a highly complex technology environment with a large volume of ongoing changes. Thousands of best practices are ever-evolving. In this kind of environment, it's almost impossible to manually identify risks, and downtime and data loss are a natural result.

Coral's automated analyses are critical for you to ensure the highest levels of availability.

With Coral's powerful risk detection engine, you can run analyses and gain visibility of your environment using deep knowledge and machine learning algorithms. And as a result, you'll be able to identify risks before they impact your business. Coral is:

- Secure
- Non-intrusive
- Proactive

What Project Coral does for you

Coral looks for potential misconfigurations and deviations from vendor and industry best practices across all layers such as virtual machines, containers, networks, load balancers, databases, cloud storage, DNS, and more.
Coral's built-in risk detection engine continuously checks for hundreds of known misconfigurations that can cause outages or even data loss. When a risk is discovered, you'll get a description of the problem, impact and the suggested resolution.

How to use this report

Assess your configuration and make improvements fast, using the resulting HealthCheck report. You'll get this information:

- Detailed description of each risk
- Step-by-step suggestions to resolve the risk

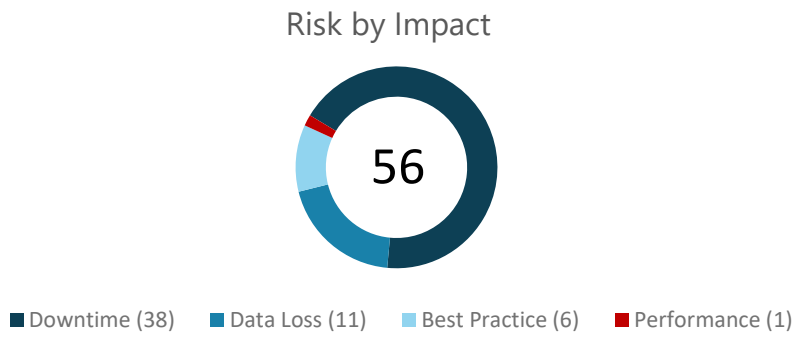
Executive summary

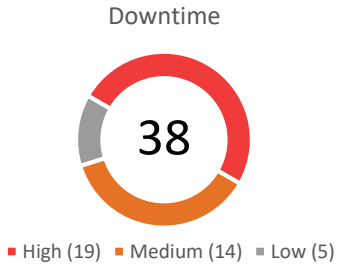
Scanned environment

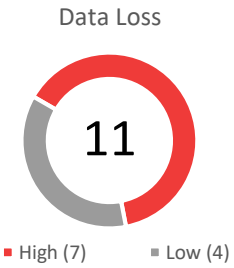
- To analyze configuration and identify trends, metadata was collected for over two weeks. There was no performance impact and no software was installed.
- A representative subset of the environment was scanned:
  - 1 AWS account including 380 EC2 instances, 27 load balancers and 11 RDS databases
- The following business services were impacted: Payments application, DWH application, M&O service, Web service, Online Banking application, CRM application.

HealthCheck findings

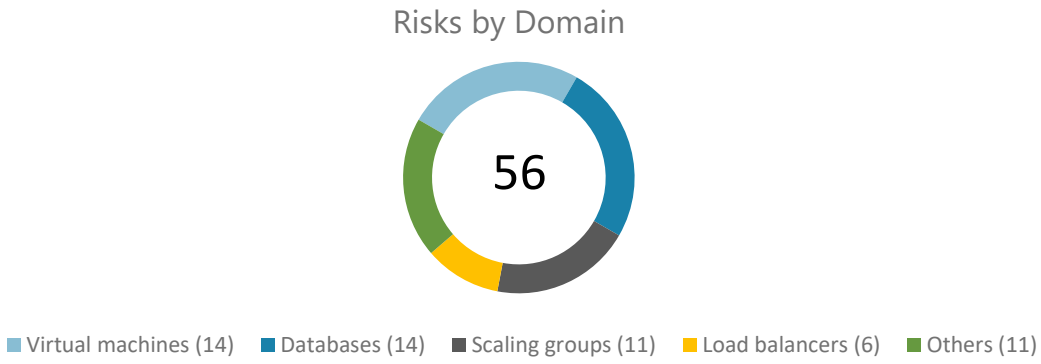
- This report contains a single risk sample for each check violation (including full details, impact and guidelines for remediation).
- 56 configuration risks were found, resulting from 26 different check violations (see Summary of check violations).
- Significant **downtime** and **data loss** configuration risks were identified, affecting the resilience of the AWS environment.







› The downtime and data loss risks span all layers of the infrastructure, including **virtual machines, databases, scaling groups, load balancers** and more.



Conclusions

› The risks found are practically impossible to manually detect. However, with Coral risk detection and recommended resolution, you can easily repair these risks. Then, your fixes are verified during the next scan.
› It is highly likely that there are additional risks in the unscanned portion of the production and staging environments.
› To proactively detect risks and ensure continuous protection, we highly recommend you use the Coral platform continuously on all production and staging environments.

Summary of check violations

26 check violations were detected, resulting in 56 separate risks:

ID	Check name	# of risks	Impact	Urgency	Domain
1	Partial VM snapshot	2	Data Loss	High	EC2
2	Non-protected EC2 with delete-on-termination	3	Data Loss	High	EC2
3	Single availability zone ECS cluster	2	Downtime	High	ECS
4	ASG with unavailable resources	3	Downtime	High	ASG
5	Auto scale group SPoF	2	Downtime	High	ASG
6	ASG with suspended processes	2	Downtime	High	ASG
7	Internet facing LB with blocked listener port	3	Downtime	High	LB
8	Internet facing LB with private subnet	1	Downtime	High	LB
9	Load balancer target instances SPoF	1	Downtime	High	LB
10	Single availability zone RDS	3	Downtime	High	RDS
11	RDS without manual snapshot	2	Data Loss	High	RDS
12	Used expired certificates	2	Downtime	High	ACM
13	EC2 service limits too low	4	Downtime	Medium	EC2
14	EBS service limits too low	1	Downtime	Medium	EC2
15	LB targets with unrestricted network access	1	Downtime	Medium	LB
16	RDS without low storage event subscription	2	Downtime	Medium	RDS
17	Route 53 failover records with high TTL	2	Downtime	Medium	Route 53
18	Route 53 failover sets with same health check	2	Downtime	Medium	Route 53
19	DynamoDB capacity limit is too low	1	Downtime	Medium	DynamoDB
20	DynamoDB configured read capacity is too low	1	Downtime	Medium	DynamoDB
21	RDS without storage encryption	2	Downtime	Low	RDS
22	RDS with default parameter group	3	Downtime	Low	RDS
23	Old snapshots	4	Data Loss	Low	EC2
24	ASG with a wrong health check type	4	Best Practice	Low	ASG
25	CloudFront custom error responses	2	Best Practice	Low	CloudFront
26	Public S3 bucket is used as a CloudFront origin	1	Performance	Low	S3
	Total	56			

Detected risks in detail (sample)

Risk urgency **High**

Risk of
Data Loss

Risk 1

Name	Partial VM snapshot
Impact	Data Loss

Summary

EC2 instance **crpqsql01** in **N.Virginia** has a partial point-in-time copy.

Description

EC2 instance **crpqsql01** in **N.Virginia** has one snapshot copy that does not contain all the EC2 volumes. This prevents restoration of the virtual machine when needed (see Impact).

The following table shows the partial copy:

VM physical volume	Volume id	Attach time	Latest snapshot date
/dev/xvda	vol-63add23b	Jan 25, 2018	Mar 30, 2019 9:00 AM
/dev/sdb	vol-ca521e54	Jan 25, 2018	Mar 30, 2019 9:00 AM
/dev/sdc	vol-0aa3a41cddbb123bc	Feb 28, 2019	N/A

Impact

Incomplete snapshots can often prevent recovery of the VM. If you experience a disaster, a security-related incident, or other problems that affect the VM, and you can't recover the VM, **significant data loss** is likely.
At times, partial copies are leftovers from projects such as maintenance or a data migration. In these cases, there is no data loss risk, but you have an opportunity to cut costs.

Affected business entities:

M&O service

Resolution

Add the missing disks to any future snapshot taken for the VM. For automated snapshots, update relevant scripts, processes or tools. You may need to take a new, complete snapshot, and then delete all the incomplete copies.

Risk urgency **High**

Risk of
Data Loss

Risk 2

Name	Non-protected EC2 with delete-on-termination
Impact	Data Loss

Summary

Static EC2 instances in **N.Virginia** with delete-on-termination don't have termination protection.

Description

There are static EC2 instances in **N.Virginia** with delete-on-termination that don't have termination protection. This configuration is very risky: An accidental instance termination through the console, the API, or the CLI can cause downtime and even data loss.

The following table shows the instances at risk:

EC2 id	EC2 name	Launch time
i-0124ffc1256ccb123	db-mysql-30ba	Aug 02 12:10 2018
i-0147468ddbbaac312	db-mongo-backup12	Dec 20 07:32 2018
i-0422dd3456cc1aac1	poc-arm22	Jan 28 15:01 2019

Impact

Without termination protection enabled, there is a risk of accidentally terminating EC2 instances. This termination can lead to downtime.
In addition, there is a risk of data loss when the **delete-on-termination** attribute is set to **true**. When this attribute is enabled, the volumes associated with the EC2 instance are deleted when the instance is terminated.

Affected business entities:

M&O service

Resolution

Make sure that static EC2 instances with delete-on-termination that are provisioned outside an auto-scaling group have the termination protection safety feature enabled. This protects the EC2 instances from accidental termination and possible downtime and data loss.

To prevent data loss during termination, set the **delete-on-termination** attribute to **false.**

To enable the termination protection feature using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **INSTANCES**, choose **instances**.
3. Select the EC2 instance to protect against accidental termination.
4. Click the **Actions** button, and then select **Instance Settings** > **Change Termination Protection**.
5. In the Enable Termination Protection dialog box, click **Yes, Enable**.

Risk urgency **High**

Risk of Downtime

Risk 3

Name	Single availability zone ECS cluster
Category	Downtime

Summary

ECS clusters in **N.Virginia** utilize a single availability zone.

Description

ECS clusters in **N.Virginia** utilize a single availability zone. This could lead to unplanned downtime during an availability zone outage (see Impact).

The following table shows the single availability zone ECS clusters:

ECS Cluster name	Auto scaling group name	Availability zone	container instances #
ecsprod	EC2ContainerService-ecsprod	us-east-1b	6
crp-po-cluster	EC2ContainerService-crp-po-cluster	us-east-1b	4

Impact

When an availability zone becomes unhealthy or unavailable, the auto scaling group launches new EC2 instances in an unaffected availability zone. If the auto scaling group used by the ECS cluster only utilizes one availability zone, it can't launch new instances. This can lead to application downtime.

Affected business entities:

CRM application

Resolution

To expand the availability of your ECS cluster, make sure that the ECS cluster's auto scaling group spans multiple availability zones within a region.

To add new availability zones to an auto scaling group using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **AUTO SCALING**, choose **Auto Scaling Groups**.
3. Select the auto scaling group to reconfigure.
4. Select the **Details** tab and click the **Edit** button.
5. In the **Subnet(s)** box, add the subnets that correspond to the new availability zones.
6. Click **Save**.

Risk urgency **High**

Risk of Downtime

Risk 4

Name	ASG with unavailable resources
Impact	Downtime

Summary

The **awseb-asg** auto scaling group points to unavailable resources.

Description

The **awseb-aslc** launch configuration of the **awseb-asg** auto-scaling group points to unavailable resources. When needed, the scaling group will not be able to spawn new EC2 instances and the scale-up will fail. This could lead to major performance issues and unplanned downtime (see Impact).

The following table shows the scaling group resources and their status:

Resource name	Resource type	Is available?
ami-c2b885a8	AMI ID	Yes
awseb-sg-1ar	Security Group	Yes
snap-0295d563	Block Device Snapshot	No
snap-16680b77	Block Device Snapshot	Yes

Impact

Scaling groups may scale up automatically when needed. If launch configuration resources are not available, new EC2 instances will not start and scaling up will fail. As a result, the application will not have sufficient resources to handle the load which will affect performance and could eventually cause downtime.

Impacted business entities:

DWH application

Resolution

Make sure all the resources in the launch configuration are available.
To prevent confusion or resource deletion:

1. Name and tag the resources used in the launch configuration correctly.
2. Never reference a shared/public image in launch configuration – select only "Owned by me" AMIs.

Risk urgency **High**

Risk of Downtime

Risk 5

Name	Auto scale group SPoF
Category	Downtime

Summary

Auto scaling groups in **N.Virginia** have a single point of failure.

Description

Auto scaling groups in **N.Virginia** have a single point of failure for one or both of these reasons:

- They utilize a single availability zone.
- They are configured with a min capacity of 1.

This could lead to unplanned downtime during an availability zone outage (see Impact).

The following table shows the auto scaling groups with SPoF:

Auto scaling group name	Desired capacity	Min capacity	Availability zones
awseb-asg	12	1	us-east-1b
crmprd-asg	4	1	us-east-1b

Impact

When an availability zone becomes unhealthy or unavailable, the auto scaling group launches new EC2 instances in an unaffected availability zone. If the auto scaling group utilizes only one availability zone, it can't launch new instances.
In addition, if the min capacity of the scale group is 1, it could scale down to a single instance which presents a single point of failure. This can lead to application downtime.

Affected business entities:

DWH application

Resolution

To expand the availability of your auto scaled applications, make sure that:

- The auto scaling groups span multiple availability zones within a region.
- The min capacity is greater than 1.

To add new availability zones to an auto scaling group using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **AUTO SCALING**, choose **Auto Scaling Groups**.
3. Select the auto scaling group to reconfigure.
4. Select the **Details** tab and click the **Edit** button.
5. In the **Min** box, make sure the min capacity is greater than 1.
6. In the **Subnet(s)** box, make sure the subnets correspond to at least two availability zones.
7. Click **Save**.

Risk urgency **High**

Risk of Downtime

Risk 6

Name	ASG with suspended processes
Impact	Downtime

Summary

The **awseb-asg** auto scaling group has suspended processes.

Description

The **awseb-asg** auto scaling group has suspended processes. When needed, the scaling group might not function correctly. This could lead to major performance issues and unplanned downtime (see Impact).

Suspended processes:

- Launch
- ReplaceUnhealthy

Impact

Scaling groups may scale up automatically when needed (usually at peak time when load is high). If one or more scaling group processes are suspended, the group behavior changes and results in unpredictable performance issues and even downtime.

Impacted business entities:

DWH application

Scaling group processes:

- **Launch** – Adds a new EC2 instance to the group, increasing its capacity.
  - If Launch is suspended, this will disrupt other processes. For example, an instance in a standby state cannot be returned to service if the Launch process is suspended since the group cannot scale.
- **Terminate** - Removes an EC2 instance from the group, decreasing its capacity.
  - If Terminate is suspended, this will disrupt other processes.
- **HealthCheck** - Checks the health of the instances.
- **ReplaceUnhealthy** - Terminates instances that are marked as unhealthy and later creates new instances to replace them.
- **AZRebalance** - Balances the number of EC2 instances in the group across the availability zones in the region.
  - If suspended and a scale out or scale in event occurs, the scaling process still tries to balance the availability zones. For example, during scale out, it launches the instance in the availability zone with the fewest instances.
  - If the Launch process is suspended, AZRebalance neither launches new instances nor terminates existing instances. If the Terminate process is suspended, the auto scaling group can grow up to ten percent larger than its maximum size.
- **AlarmNotification** - Accepts notifications from CloudWatch alarms that are associated with the group.
  - If suspended, Amazon EC2 auto scaling does not automatically execute policies that would be triggered by an alarm. If Launch or Terminate are suspended, it would not be able to execute scale out or scale in policies, respectively.
- **ScheduledActions** - Performs scheduled actions that you create.
  - If Launch or Terminate are suspended, scheduled actions that involve launching or terminating instances are affected.

- **AddToLoadBalancer** – Adds instances to the attached load balancer or target group when they are launched.
  - If suspended, Amazon EC2 auto scaling launches the instances but does not add them to the load balancer or target group. If the AddToLoadBalancer process is resumed, it resumes adding instances to the load balancer or target group when they are launched. However, it does not add the instances that were launched while this process was suspended. Those instances must be registered manually.

Resolution

Resume the suspended processes using the AWS Management Console.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **AUTO SCALING**, choose **Auto Scaling Groups**.
3. Select the auto scaling group that you want to reconfigure.
4. Select the **Details** tab and click the **Edit** button.
5. Remove all the processes from the **Suspended Processes** list.
6. Click **Save**.

Risk urgency **High**

Risk of Downtime

Risk 7

Name	Internet facing LB with blocked listener port
Impact	Downtime

Summary

Internet-facing load balancer **aws-lb-remote-dim** in **N.Virginia** has a blocked listener port.

Description

Internet-facing load balancer **aws-lb-remote-dim** in **N.Virginia** has a listener port that is blocked by a security group configuration. This could lead to application availability issues (see Impact).

Details:

- Load balancer: aws-lb-remote-dim
- Load balancer type: application load balancer
- Listener ports: HTTP/80, TCP/443
- Security groups: sg-d2341bc1
- Listener ports blocked by SGs: 443

Impact

When the listening port of a load balancer is blocked by a security group, incoming traffic to the load balancer on that port is dropped. This can lead to application downtime.

Impacted business entities:

Web service

Resolution

To allow incoming traffic to the load balancer of the blocked port using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Select the load balancer that you want to update.
4. In the **Description** tab find the **Source Security Group** and click it.
5. Go to the security group **Inbound** tab and click **Edit**.
6. Add the missing rule. For example:
    a. Type: Custom TCP Rule
    b. Port range: <the blocked port>
    c. Source: Anywhere
7. Click **Save**.

Risk urgency **High**

Risk of Downtime

Risk 8

Name	Internet facing LB with private subnet
Impact	Downtime

Summary

Internet-facing load balancer **aws-lb-remote-dim** in **N.Virginia** is configured in a private subnet.

Description

Internet-facing load balancer **aws-lb-remote-dim** in **N.Virginia** is configured in a private subnet. This could lead to application availability issues (see Impact).

The following table shows the subnets of load balancer **aws-lb-remote-dim**:

Subnet id	Subnet name	Availability zone	Is public
subnet-a245d145	DIM-Public-Subnet	us-east-1a	Yes
subnet-2a18ca2c	CRM-PROD-Subnet	us-east-1b	No

Impact

When one of the subnets of an internet-facing load balancer is private -- meaning it does not have a route to an internet gateway -- incoming traffic for this subnet is dropped. This can lead to application downtime.

Impacted business entities:

Web service

Resolution

To change the subnets of a load balancer using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Select the load balancer to reconfigure.
4. Click **Actions** and select **Edit subnets**.
5. Make sure that all the subnets of the internet-facing load balancer are public.
6. Click **Save**.

Risk urgency **High**

Risk of Downtime

Risk 9

Name	Load balancer target instances SPoF
Category	Downtime

Summary

Application load balancer **aws-lb-remote-dim** in **N.Virginia** only has healthy target instances in a single availability zone.

Description

Application load balancer **aws-lb-remote-dim** in **N.Virginia** only has healthy target instances in a single availability zone. This could lead to unplanned downtime in case of an availability zone outage (see Impact).

The following table shows the instances in target group **aws-lb-remote-dim**:

Target EC2 id	Target EC2 name	Availability zone	Instance state
i-04562645aaa33532a	[Dim] LBRemote-01	eu-west-1a	healthy
i-a357a3ff35bb35a3c	[Dim] LBRemote-02	eu-west-1a	healthy
i-5674acc3464bb1aa2	[Dim] LBRemote-03	eu-west-1a	healthy
i-124a24512a4d46a21	[Dim] LBRemote-04	eu-west-1b	unhealthy

Impact

When an availability zone becomes unhealthy or unavailable, the load balancer can use the target instances in unaffected availability zones. If the load balancer has healthy target instances in a single availability zone and that availability zone becomes unhealthy, the load balancer stops redirecting traffic. This can lead to application downtime.

Impacted business entities:

Web service

Resolution

Make sure there are healthy target instances in at least two availability zones.

Risk urgency **High**

Risk of Downtime

Risk 10

Name	Single availability zone RDS
Category	Downtime

Summary

RDS instances in **N.Virginia** are deployed in a single availability zone.

Description

RDS instances in **N.Virginia** are deployed in a single availability zone. This could lead to unplanned downtime during a database outage (see Impact).

The following table shows the single availability zone RDS instances:

DB identifier	Engine	Size	Availability zone	Create time
crmasses	mysql	db.m5.large	us-east-1a	Mar 26, 2017
dimrmtdb	postgres	db.t3.large	us-east-1a	Jan 14, 2018

Impact

Amazon RDS with multi-AZ deployment maintains a synchronous standby replica in a different availability zone from the active DB instance. This configuration provides data redundancy, high availability and failover support. It also eliminates I/O freezes and minimizes latency spikes during system backups. RDS configuration without multi-AZ deployment is vulnerable to outages that could lead to application downtime.

Resolution

Modify the RDS database instance to enable multi-AZ deployment.

Use the following steps to enable multi-AZ RDS deployment:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/rds.
2. Click **Instances**.
3. Click on the instance to reconfigure.
4. Click **Instance actions** and **Modify**.
5. Select **yes** in multi-AZ deployment.
6. Click **Continue** and **Modify DB Instance**.

Risk urgency **High**

Risk of
Data loss

Risk 11

Name	RDS without manual snapshot
Category	Data loss

Summary

RDS instances in **N.Virginia** have no manual snapshots.

Description

RDS instances in **N.Virginia** have no manual snapshots. An accidental deletion of the table can lead data loss (see Impact).

The following table shows the RDS instances without a manual snapshot:

DB identifier	Engine	Size	Availability zone	Create time
crmasses	mysql	db.m5.large	us-east-1a	Mar 26, 2017
dimrmtdb	postgres	db.t3.large	us-east-1a	Jan 14, 2018

Impact

When an RDS instance is deleted, all the automatic snapshots are deleted as well. If there is no manual snapshot, the data will not be recoverable. This means that an accidental deletion of the instance results in having no snapshots and recovery is not possible.

Resolution

1. To create a manual backup using the console:
2. Open the Amazon EC2 console at https://console.aws.amazon.com/rds.
3. Click **Databases**.
4. Click on the database you wish to reconfigure.
5. Click the **Maintenance & Backups** tab.
6. Click **Take snapshot**.
7. Enter the snapshot name and click **Take Snapshot**.

Risk urgency **High**

Risk of Downtime

Risk 12

Name	Used expired certificates
Category	Downtime

Summary

There are used expired certificates in region **N.Virginia**.

Description

There are used expired certificates in **N.Virginia**. This could lead to availability issues if the certificates are used by client facing applications (see Impact).

The following table shows the expired certificates:

Certificate domain name	Type	Expiration date	Used by
*.dtmservice.io	Imported	Jun 15, 2019	loadbalancer/ aws-lb-remote-dim

Impact

SSL certificates facilitate the encryption of data in transit. The SSL certificates allow you to use HTTPS to create secure, encrypted connections. An expired SSL certificate on a website can drive users away from using it.
In addition, an expired SSL certificate can lead to phishing scams where the victims (the users of the web site) are lured into giving their personal information, which is misused later by cyber attackers.

Impacted business entities:

Payments application

Resolution

Renew the expired certificates or delete them if they are not needed.

Risk urgency **Medium**

Risk of Downtime

Risk 13

Name	EC2 service limits too low
Impact	Downtime

Summary

The EC2 service limit of instance type "**m1.large**" in **N.Virginia** is too low.

Description

The EC2 service limit of instance type "**m1.large**" in **N.Virginia** is **32**. This limit is too low, since the aggregated maximum size of all scaling groups is **48**. This could lead to unplanned downtime when groups scale up at the same time (see Impact).

The following table shows the scaling groups containing m1.large instances and their capacity:

Scaling group	Max size	Current capacity
awseb-M1-SG	10	4
awsewd-43	10	4
prctrl-ds-SG	10	4
crporc-SG	6	4
"Static instances"	12	12
Total	**48**	**28**

Impact

Scaling groups scale up automatically as needed. At peak times, when load is high, multiple scaling groups may require more resources at the same time.

When the service limit is reached, new EC2 instances will not start, and scaling up will fail. As a result, the application has insufficient resources to handle the load. This affects performance and could eventually cause downtime.

Resolution

Use the Limits page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2. Make a separate request for each region.

To request a limit increase:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a region.
3. From the navigation pane, choose **Limits**.
4. Find the instance type in the list. Choose **Request limit increase**.
5. Complete the limit increase form. Amazon will respond to your request.

Risk urgency **Medium**

Risk of Downtime

Risk 14

Name	EBS service limits too low
Impact	Downtime

Summary

The EBS volumes size limit of "**General Purpose SSD**" (gp2) in **Ohio** is too low.

Description

The EBS volumes size limit of "**General Purpose SSD**" (gp2) in **Ohio** is **307200** GiB. This limit is too low, since the aggregated maximum size of all scaling groups is **488120** GiB. This could lead to unplanned downtime when groups scale up at the same time (see Impact).

The following table shows the scaling groups containing gp2 volumes and their size:

Scaling group	Max size (GiB)	Current size (GiB)
database-mgmt-sg	100000	50000
cluster-mgmt-sg	100000	15200
masters-mgmt-sg	100000	64850
nodes-mgmt-sg	100000	73174
"Static instances"	88120	88120
Total	**488120**	**291344**

Impact

Scaling groups scale up automatically as needed. At peak times, when load is high, multiple scaling groups may require more resources at the same time. When the EBS volume storage limit is reached, new volumes will not be created. This causes a failure to spawn new EC2 instances, and the scaling up will fail. As a result, the application has insufficient resources to handle the load. This lack of resources affects performance and could eventually cause downtime.

Resolution

Use the Limits page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2. This should be done separately for each region.

To request a limit increase:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a region.
3. From the navigation pane, choose **Limits**.
4. Locate the EBS limit in the list. Choose **Request limit increase**.
5. Complete the limit increase form. Amazon will respond to your request.

Risk urgency **Medium**

Risk of Downtime

Risk 15

Name	LB targets with unrestricted network access
Impact	Downtime

Summary

Target EC2 instances of application load balancers in **N.Virginia** are exposed to the internet.

Description

Target EC2 instances of application load balancers in **N.Virginia** are exposed to the internet. Applications that access a target directly may suffer unplanned downtime on a single target EC2 instance failure (see Impact).

The following table shows exposed target EC2 instances:

Load balancer	Target EC2 instances	Security group	Open inbound permissions
aws-lb-remote-dim	i-00aa23a456bb1bb24 i-04134aa2ee4abcc3a	sg-0d2eb1b451f43a241	tcp/80 tcp/443
aws-lb-crm-prd	i-23a34ee35a5b5c32a i-0c2af3ca346aab3ef	sg-0cca231ab3451a358	tcp/80

Impact

If the load balancer target EC2 instances are exposed to the internet, applications can access the targets directly, without going through the load balancer. Applications that are configured to access the load balancer target EC2 instances directly will lose the target resiliency that the load balancer provides. This means that if one of the load balancer's target EC2 instances fail, the application might suffer from downtime.

Impacted business entities:

Web service

Resolution

Make sure that the target EC2 instances are accessible only to the load balancer and that all the traffic goes through the load balancer.

In addition, we recommend separating the load balancer from the target instances: place the load balancer in a public subnet and the target instances in a different, private subnet.

Risk urgency **Medium**

Risk of Downtime

Risk 16

Name	RDS without low storage event subscription
Impact	Downtime

Summary

RDS instances in **N.Virginia** have no event subscription for low storage.

Description

RDS instances in **N.Virginia** have no event subscription for low storage. This could lead to unplanned downtime if the database runs out of storage space (see Impact).

The following table shows the RDS instances without low storage event subscription:

DB identifier	Engine	Size	Create time
crmasses	mysql	db.m5.large	Mar 26, 2017
dimrmtdb	postgres	db.t3.large	Jan 14, 2018

Impact

Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud. For production workloads, we strongly recommend using event notifications to proactively address performance issues and outages. The "low storage" notification sends an alert when the database instance has consumed more than 90% of its allocated storage.

Resolution

Subscribe the databases to the low storage RDS event notification:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/rds.
2. In the navigation pane, choose **Event Subscriptions**.
3. In the Event Subscriptions pane, choose **Create Event Subscription**.
4. In the Create Event Subscription dialog box:
    a. Next to **Name**, type a name for the event notification subscription.
    b. For **Send Notifications To**, choose an existing Amazon SNS Amazon Resource Name (ARN) for an Amazon SNS topic, or choose Create Topic to enter the name of a topic and a list of recipients.
    c. For **Source type**, choose **Instances**.
    d. For **instances to include**, choose either All Instances or the specific RDS instance.
    e. For **event categories to include**, choose **low storage**.
    f. Click **Create**.

Risk urgency **Medium**

Risk of Downtime

Risk 17

Name	Route 53 failover records with high TTL
Impact	Downtime

Summary

Route 53 failover records in hosted zone **dtmprod.io.** are configured with a high Time to Live (TTL) value.

Description

Route 53 failover records in hosted zone **dtmprod.io.** are configured with a high TTL value. This prevents clients from rapidly using the secondary record in case of a failover. A too-high TTL value can cause application downtime (see Impact).

The following table shows the failover records with high TTL values:

Hosted zone id	Hosted zone name	Failover record name	TTL
Z997E45AMDD3B1	dtmprod.io.	mysql-remote.dtmprod.io	300
Z997E45AMDD3B1	dtmprod.io.	secure.dtmprod.io	300

Impact

TTL is the number of seconds a DNS resolver caches a response. The value is associated with every record. AWS recommends a TTL of 60 seconds or less when using DNS Failover, to minimize the amount of time it takes to stop routing traffic to your failed endpoint.

Impacted business entities:

Online Banking application

Resolution

To update the TTL of the Route 53 failover records using the console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/route53/.
2. Click **Hosted Zones**.
3. Select the hosted zone that contains the failover records.
4. Select the failover record to update.
5. Change the **TTL** to the desired value (in seconds).
6. Click **Save Record Set**.

Risk urgency **Medium**

Risk of Downtime

Risk 18

Name	Route 53 failover sets with same health check
Impact	Downtime

Summary

The **cred.dtmprod.io.** Route 53 failover record set uses the same health check for both the primary and secondary resource record sets.

Description

The **cred.dtmprod.io.** Route 53 failover record set uses the same health check for both the primary and secondary resource record sets. This could lead to unplanned downtime in case of a DNS failover (see Impact).

Additional info:

- Hosted zone name: **dtmprod.io.**
- Hosted zone id: **Z997E45AMDD3B1**
- Resource record set type: **A**
- Resource record set name: **cred.dtmprod.io.**

Impact

Route 53 failover is used when more than one resource is performing the same function. It checks the health of the resources and responds to DNS queries using only the healthy resources. If the same health check is configured both to the primary and secondary record sets, the failover will fail.

Impacted business entities:

Online Banking application

Resolution

Create separate health checks for the primary and secondary resource record sets.

To update the Health Check of a record set using the Route 53 console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/route53/.
2. Click **Hosted Zones**.
3. Select the hosted zone that contains the failover records.
4. Click the row for the record that you want to edit.
5. Set the **Health Check to Associate** to the correct health check.
6. Click **Save Record Set**.

Risk urgency **Medium**

Risk of Downtime

Risk 19

Name	DynamoDB capacity limit is too low
Impact	Downtime

Summary

The DynamoDB capacity limit in **N.Virginia** is too low.

Description

The DynamoDB read and write capacity limit in **N.Virginia** is **80000**.This limit is too low, since the aggregated max capacity of all the tables higher. This could lead to unplanned downtime when several tables capacity scales up at the same time (see Impact).

The following table shows the tables and their max read and write capacity:

Table name	Read capacity	Write capacity	Max read capacity	Max write capacity
dtm-bi-data	5	5	40000	40000
dtm-mmr-emr	500	500	40000	40000
dtm-lock-status	500	100	40000	40000
Total	1005	605	120000	120000

Impact

DynamoDB auto scaling uses the AWS application auto scaling service to automatically and dynamically adjust provisioned throughput capacity, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic without throttling.

At peak times, when load is high, multiple tables may require more capacity at the same time. When the capacity limit is reached, the tables cannot scale up. As a result, the table may not be able to handle the load. In turn, this affects performance and could cause downtime.

Resolution

Use the AWS support center to create a support case for a DynamoDB service limits increase. Make a separate request for each region:

To request DynamoDB service limit increase:

1. Open the amazon support center at https://console.aws.amazon.com/support/.
2. Click **Create case** and select **Service limit increase**.
3. In the **Limit type**, select DynamoDB.
4. Select the appropriate region and contact options and click submit. Amazon will respond to your request.

Risk urgency **Medium**

Risk of Downtime

Risk 20

Name	DynamoDB configured read capacity is too low
Impact	Downtime

Summary

The minimum provisioned read capacity of DynamoDB **dtm-bi-data** in **N.Virginia** is too low.

Description

The minimum provisioned read capacity of DynamoDB **dtm-bi-data** in **N.Virginia** is **5.** This limit is too low. As a result, table scaling attempts fail, and this can cause performance issues and even application downtime (see Impact).

The following table shows the DynamoDB table scale attempts and failures:

Date	Scale attempts	# of failures	Failure reason
Mar 25, 2019	94	81	LimitExceededException
Mar 27, 2019	71	57	LimitExceededException

Failure reason details:

- **LimitExceededException**: Provisioned throughput changes are limited within a given UTC day. After the first 4, each subsequent change in the same UTC day can be performed at most once every 3600 seconds.

Impact

DynamoDB auto scaling uses the AWS application auto scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without throttling. When the workload decreases, application auto scaling decreases the throughput to avoid paying for unused capacity.

When the minimum provisioned capacity is too low, the application auto scaling attempts to scale the table very frequently, and this can cause scaling failures. As a result, the table might not be able to handle the load. This affects performance and could eventually cause downtime.

Resolution

To increase the minimum provisioned capacity using the console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/dynamodb/.
2. Click **Tables** in the left menu and select the table to update.
3. Click the **Capacity** tab, and then under **Auto Scaling**, increase the **Minimum provisioned capacity** for read and/or write.
4. Click **Save**.

Risk 21

Name	RDS without storage encryption
Impact	Downtime

Summary

RDS instances in **N.Virginia** do not have storage encryption.

Description

RDS instances in **N.Virginia** do not have storage encryption. This could lead to data security risks and even downtime if you enable storage encryption later (see Impact).

The following table shows the RDS instances without storage encryption:

DB identifier	Engine	Size	Create time
crmasses	mysql	db.m5.large	Mar 26, 2017
dimrmtdb	postgres	db.t3.large	Jan 14, 2018

Impact

You can encrypt your Amazon RDS DB instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instances. Data that is encrypted at rest includes the underlying storage for DB instances, its automated backups, the read replicas and snapshots. You cannot enable storage encryption if the database was created without encryption. This means that if you need to enable encryption, you will need to back up and restore your database – an action that results in a long downtime.

Resolution

Make sure that all the RDS instances are using storage encryption.

Risk urgency **Low**

Risk of
Downtime

Risk 22

Name	RDS with default parameter group
Impact	Downtime

Summary

RDS instances in **N.Virginia** are using the default parameter group.

Description

RDS instances in **N.Virginia** are using the default parameter group. This could lead to unplanned downtime if changes in parameters are needed (see Impact).

The following table shows the RDS instances using the default groups:

DB identifier	Engine	Size	Create time	Default parameter group
crmasses	mysql	db.m5.large	Mar 26, 2017	default.mysql5.7
dimrmtdb	postgres	db.t3.large	Jan 14, 2018	default.postgres11

Impact

Parameter groups are used to manage the RDS database configuration. Each RDS instance is associated to one parameter group. Using the default group is not recommended because it is read-only. This means that if you need to change a parameter, you will need to associate the instance to a different, user-defined group – an action that requires downtime.

Resolution

Make sure that each RDS instance is associated to dedicated, non-default parameter group.

Risk urgency **Low**

Risk of
Data Loss

Risk 23

Name	Old snapshots
Impact	Data Loss

Summary

The latest snapshot of EC2 instances in region **N.Virginia** is older than 6 months.

Description

The latest snapshot of EC2 instances in region **N.Virginia** is older than 6 months. If you need to restore the virtual machine, it will restore to an old snapshot and you risk data loss.

The following table shows the EC2 instances with old snapshots:

EC2 id	EC2 name	Latest snapshot date
i-0124ffc1256ccb123	db-mysql-30ba	Sep 20 11:00 2018
i-0147468ddbbaac312	db-mongo-backup12	Sep 20 15:42 2018
i-0422dd3456cc1aac1	poc-arm22	Oct 12 08:12 2018

Impact

In case you experience a disaster, a security-related incident, or other problems that affect the VM and there are no current snapshots, you will only be able to restore the EC2 to an old snapshot. As a result, you will lose all changes made to the configuration or data after the last snapshot.

Resolution

First, determine the root cause of the problem. For automated snapshots, update relevant scripts, processes or tools. If snapshots are taken manually, create a new current snapshot of the EC2.

Risk urgency **Low**

Best practice violation

Risk 24

Name	ASG with a wrong health check type
Impact	Best practice

Summary

Auto scaling groups in **N.Virginia** are not configured with the recommended health check type.

Description

Auto scaling groups in **N.Virginia** are not configured with the recommended health check type. This could lead to instance state inconsistencies between the scale group and the load balancer. Inconsistencies might cause application availability issues (see Impact).

The following table shows the blocked listener ports:

Auto scaling group name	Load balancer	Current health check type	Recommended heath check type
autoscaling-workers-02	aws-lb-workers-02	EC2	ELB
autoscaling-remote-dim	aws-lb-remote-dim	EC2	ELB
autoscaling-internal-slim	aws-lb-internal-slim	EC2	ELB

Impact

By using the right health check type, you can increase the availability of the applications deployed in these groups.

Resolution

Make sure the health check for the auto scaling group is configured correctly to detect whether the auto scaling group's registered instances are healthy. If you are using a load balancer with the auto scaling group, make sure the ELB health check is enabled. If you aren't using a load balancer, make sure the EC2 health check is enabled.

To change the heath check type using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **AUTO SCALING**, choose **Auto Scaling Groups**.
3. Select the auto scaling group to reconfigure.
4. Select the **Details** tab and click the **Edit** button.
5. Change the **Health Check Type** to the recommended setting.
6. Click **Save**.

Risk urgency **Low**

Best practice violation

Risk 25

Name	CloudFront custom error responses
Impact	Best practice

Summary

CloudFront custom error responses should be configured for all 5xx HTTP error codes.

Description

It is important to configure CloudFront custom error responses for all 5xx HTTP error codes. Without custom error responses, there is a risk of application downtime (see Impact).

The following table shows the CloudFront distributions without complete 5xx custom error responses configuration:

Distribution id	Domain name	Aliases	Custom error responses
D5637E1J75Z11C	s24b7a2ht4yr31.cloudfront.net	static-dtm-pub.dtm.io	502, 504
D3D435JJS2440D	s2xr345fds25gv.cloudfront.net	static-dtm-red.dtm.io	None

Impact

When a viewer call results in an error from the origin server, such as server is busy or unavailable, CloudFront returns the error status code to the viewer. If custom error responses are not configured, CloudFront caches the error status code for five minutes. This means that for the next five minutes, any additional identical viewer call receives the same error response, even if the error was temporary and already resolved. As a result, the application might incorrectly appear unavailable.

Impacted business entities:

Payments application

Resolution

Make sure to configure custom error responses for all 5xx HTTP error codes. We recommend configuring custom error responses for all the 4xx HTTP error codes as well.

To configure custom error responses for a CloudFront distribution using the AWS console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/cloudfront/.
2. On the **Distributions** page, select the web distribution to update.
3. Click **Distribution Settings,** and then click the **Error Pages** tab.
4. For each missing HTTP error code:
    a. Click **Create Custom Error Response**.
    b. Select the HTTP error code from the list.
    c. Enter the error caching TTL.
    d. Click **Create**.

Risk urgency **Low**

Performance Recommendation

Risk 26

Name	Public S3 bucket is used as a CloudFront origin
Impact	Performance

Summary

Public S3 bucket **media-dtm-us-east-1-s3** in **N.Virginia** is used as a CloudFront origin.

Description

Public S3 bucket **media-dtm-us-east-1-s3** in **N.Virginia** is used as a CloudFront origin. The recommended S3 ACL configuration for a CloudFront bucket origin should allow access only though CloudFront. Using a public S3 bucket as an origin presents a security risk and can also affect performance and cost (see Impact).

The following table shows the CloudFront distribution and the S3 ACL:

Distribution id	Domain name	S3 origin Public access
D5637VBB75ZWNC	s24b7a2ht4yr31.cloudfront.net	Read

Impact

Granting **Read** public access to an S3 bucket allows everyone to read the bucket objects. S3 buckets used as a CloudFront origin should be accessible only though CloudFront. This could lead to performance and cost issues.

Impacted business entities:

Payments application

Resolution

Make sure that S3 buckets used as CloudFront origins are not public.

To remove S3 public access using the AWS console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/s3/.
2. Select the S3 bucket to reconfigure, and then select the **Permissions** tab.
3. Click the **Access Control List** subtab.
4. Under **Public Access**, click the **Everyone** row and clear the checkboxes.
5. Click **Save**.