
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ENT227

How NAB improved application functionality and more by migrating to AWS

Joshua Samuel

Enterprise Solutions Architect,
Amazon Web Services

Ankit Singh

Sr. Consultant Solution Design,
National Australia Bank

Robert Hanekom

Distinguished Engineer,
National Australia Bank

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Who we are

Business objectives

Our challenges

Technical implementation

Results achieved

What’s next?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

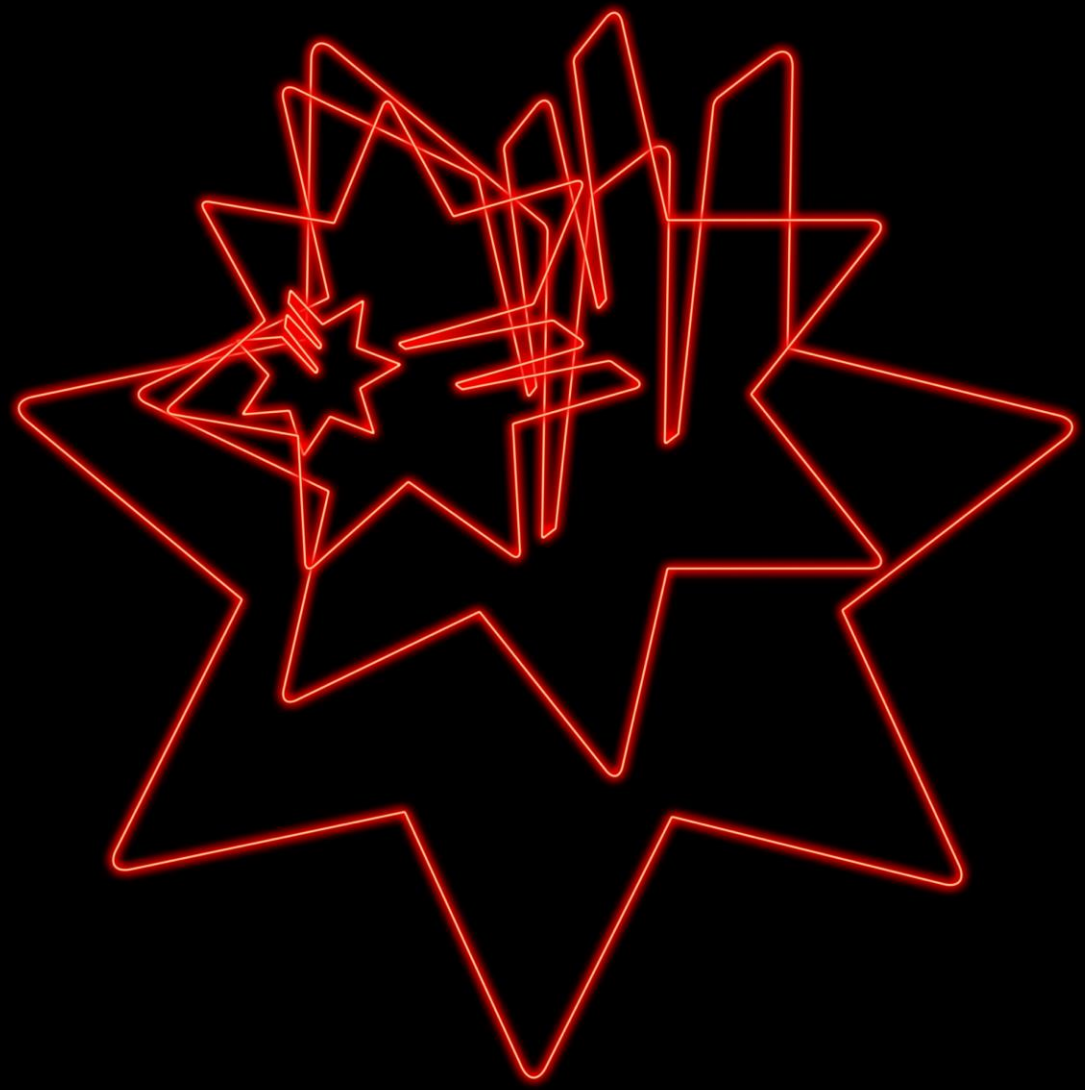
Who we are

NATIONAL AUSTRALIA BANK

32,000 colleagues

- Personal accounts
- Private clients
- Small to large businesses
- Governments

2,000+ applications

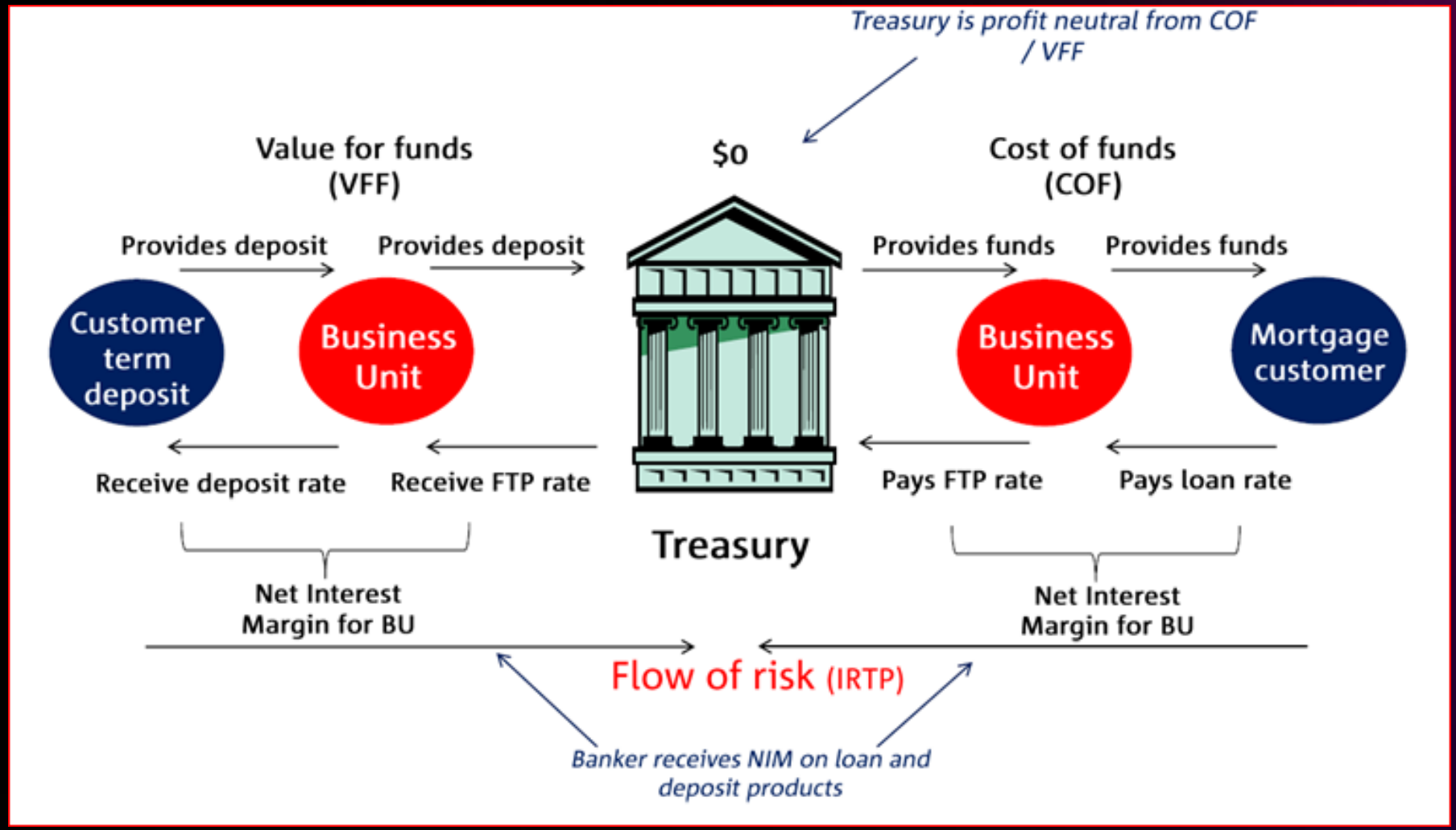


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Funds transfer pricing



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Applications we use

Oracle Financial Services Analytical Applications (OFSAA)	Oracle Funds Transfer Pricing
	Oracle Profitability Management
	Oracle GL Reconciliation
Oracle E-Business Suite	Oracle Financial Accounting Hub
	Oracle Financials General Ledger

90% of bank's balance sheet

15 million arrangements related to millions of customers with over 11 million transactions processed daily

Real-time API enabled economic cost calculation for core banking system

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Business objectives in migrating Oracle apps

Compliance risks

Out of support systems

Simplified systems

Speed to value

Robust and scalability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

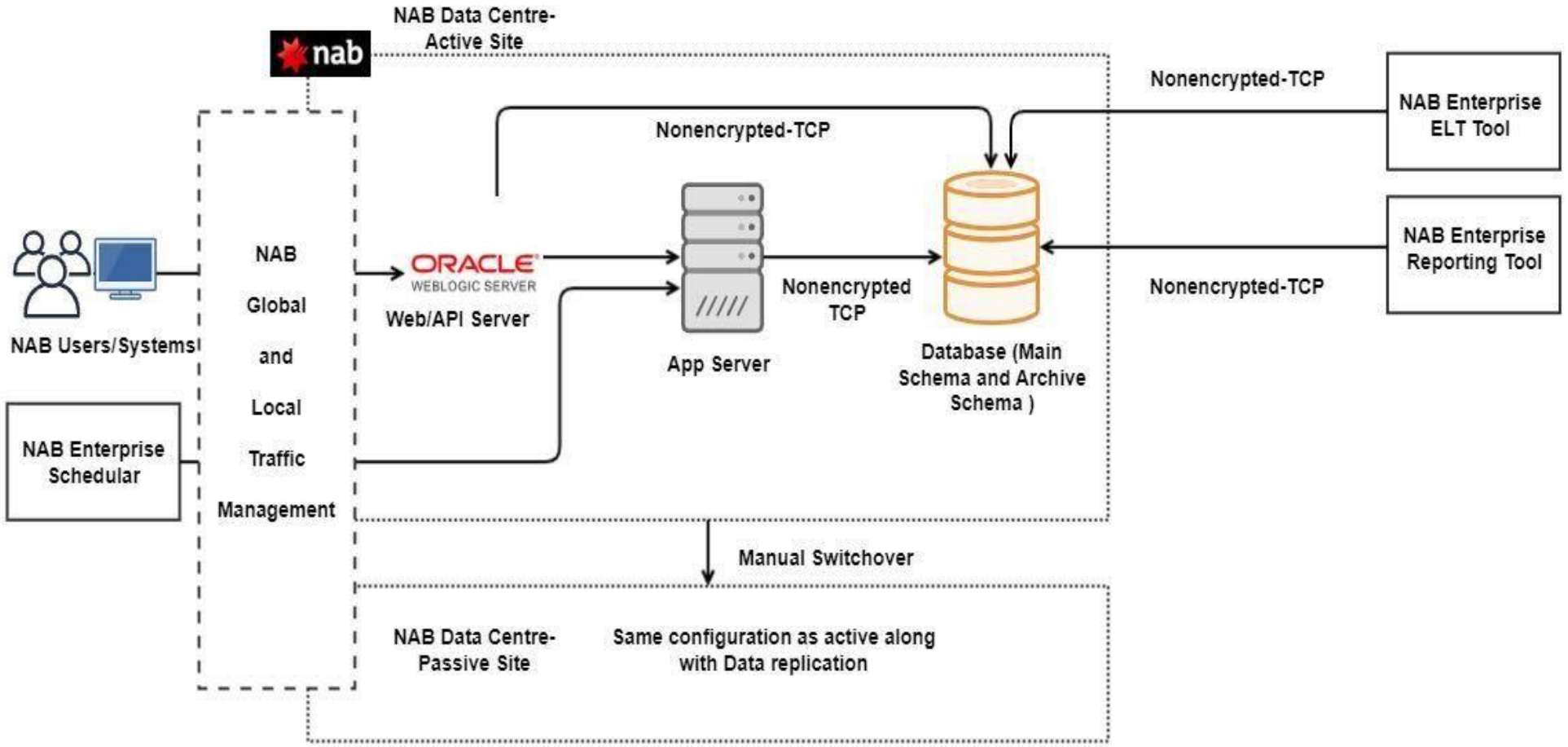
Our challenges

- Consistent software and data model for environments
- Inbuilt security, high availability, and fault tolerance baked into infrastructure automations
- Fully automated, configurable, immutable, and repeatable infrastructure
- High availability setup for self managed database for Oracle E-Business Suite app
- Performance optimization for Amazon RDS for Oracle for the OFSAA app
- Managing huge volume of data (20 TB)
- Cost optimization

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High level architecture on-prem

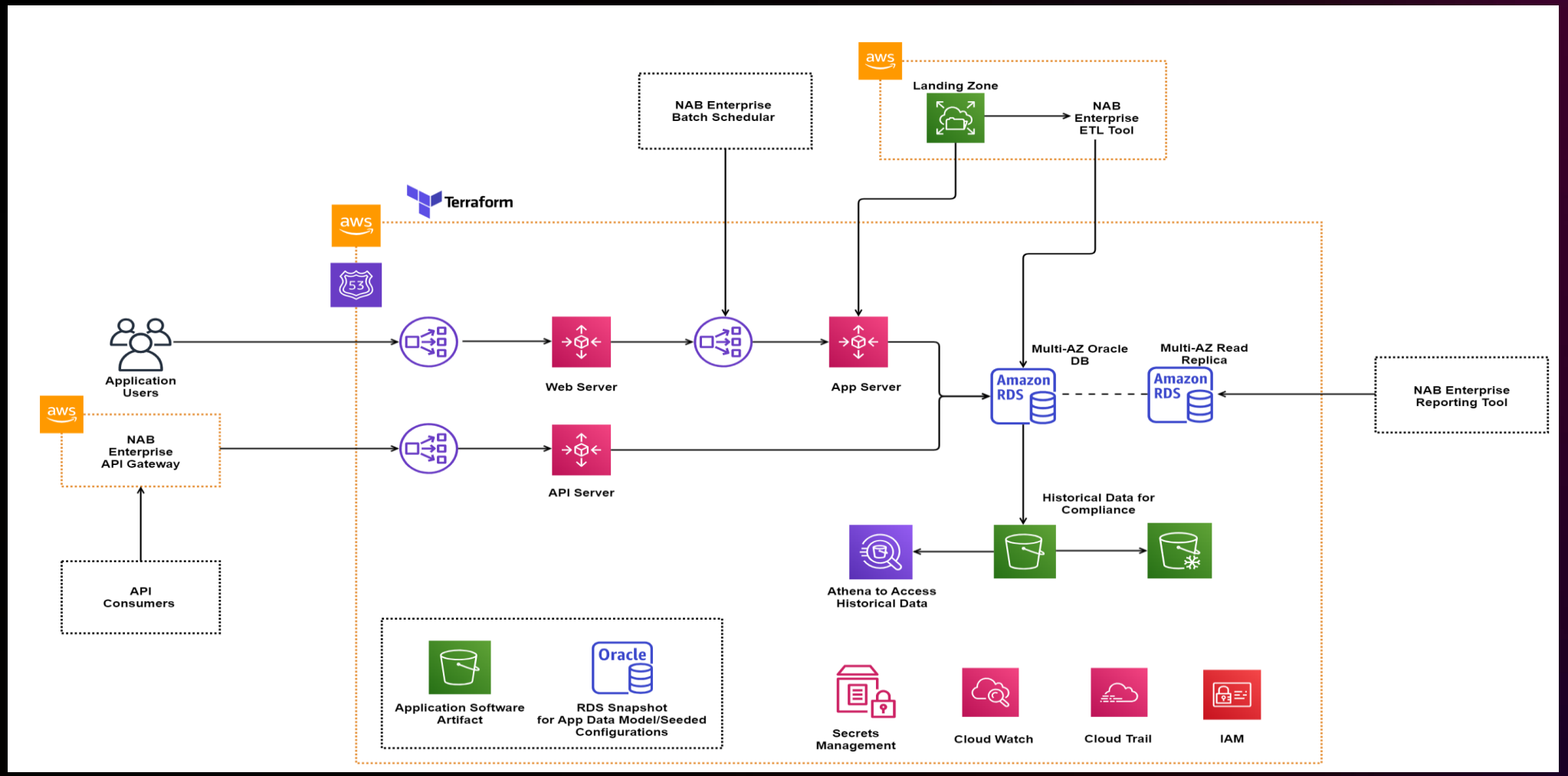
ORACLE FINANCIAL SERVICES ANALYTICS APPLICATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High level architecture AWS

ORACLE FINANCIAL SERVICES ANALYTICS APPLICATION

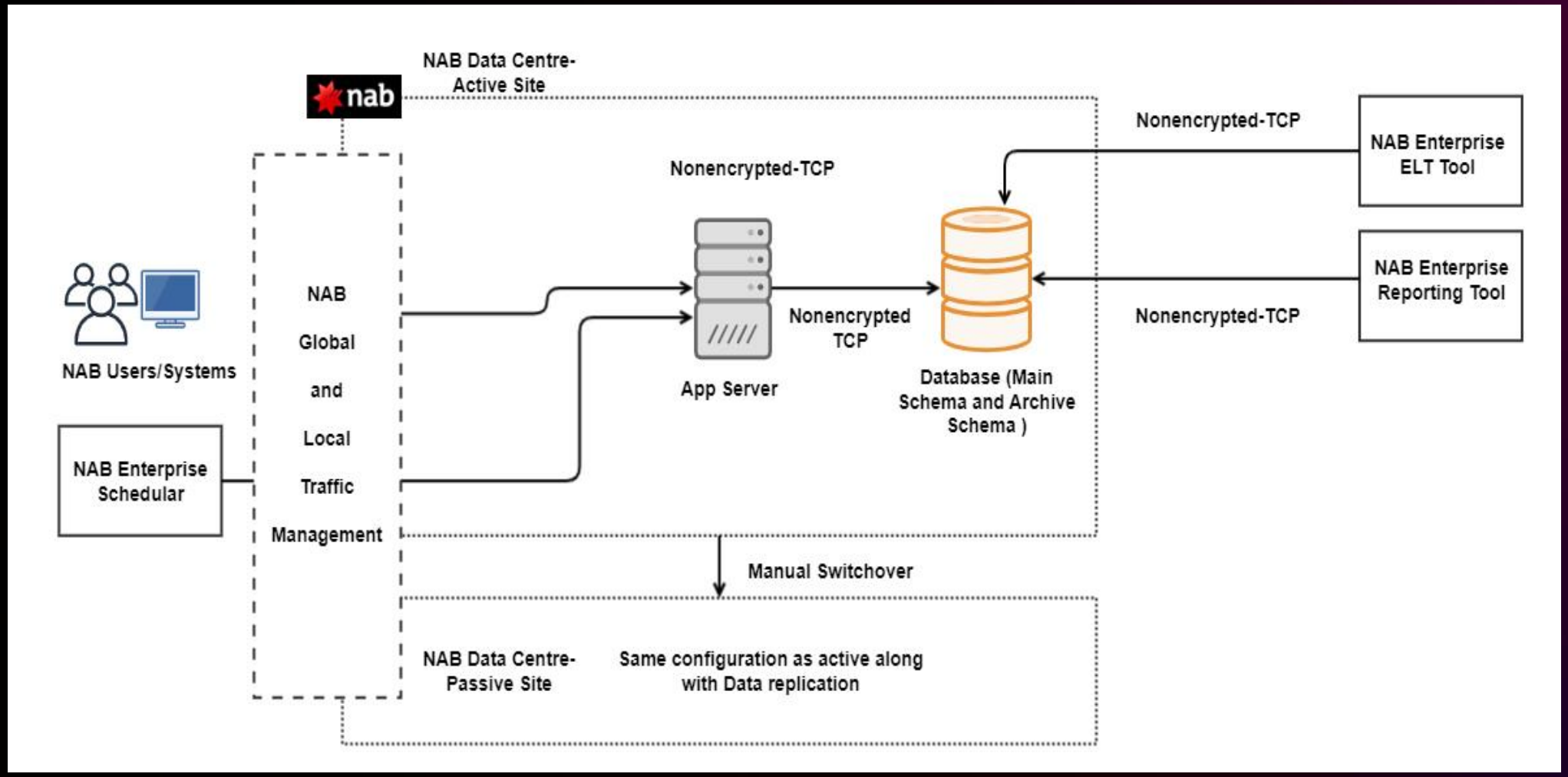


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

High level architecture on-prem

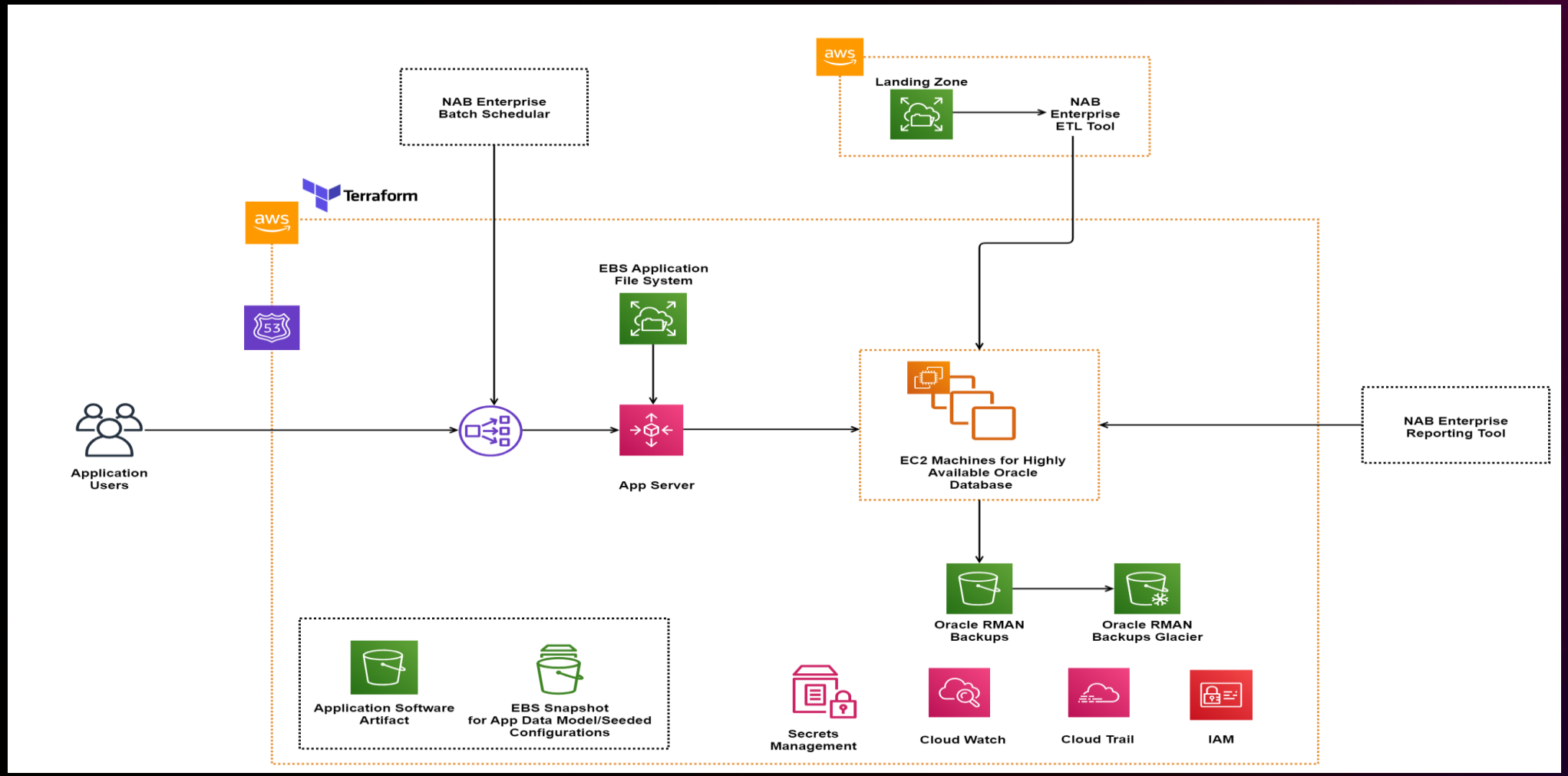
ORACLE E-BUSINESS SUITE APPLICATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High level architecture AWS

ORACLE E-BUSINESS SUITE APPLICATION

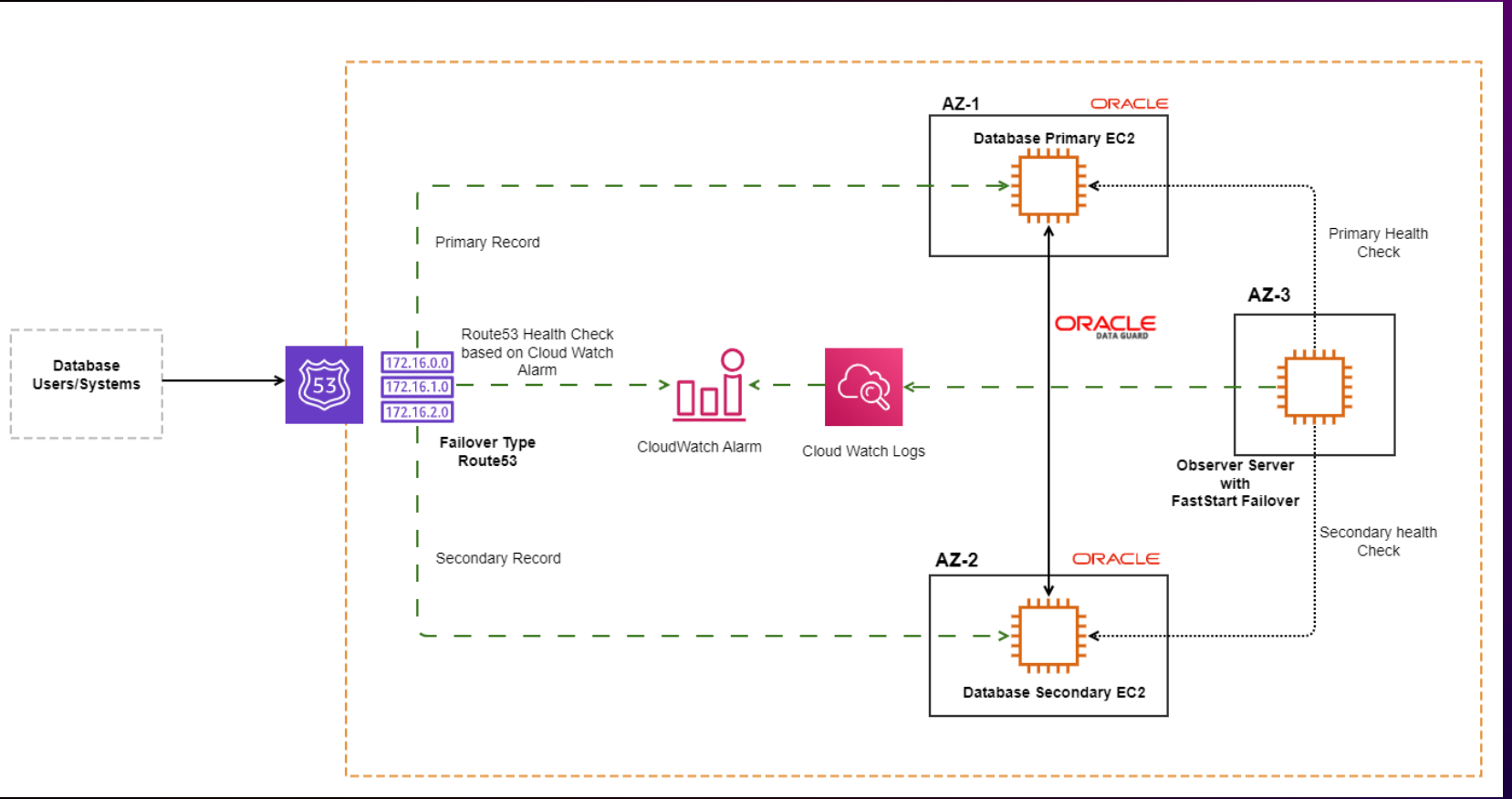


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Oracle database on Amazon EC2

- High availability
- Faster disaster recovery
- Consistent database connection string

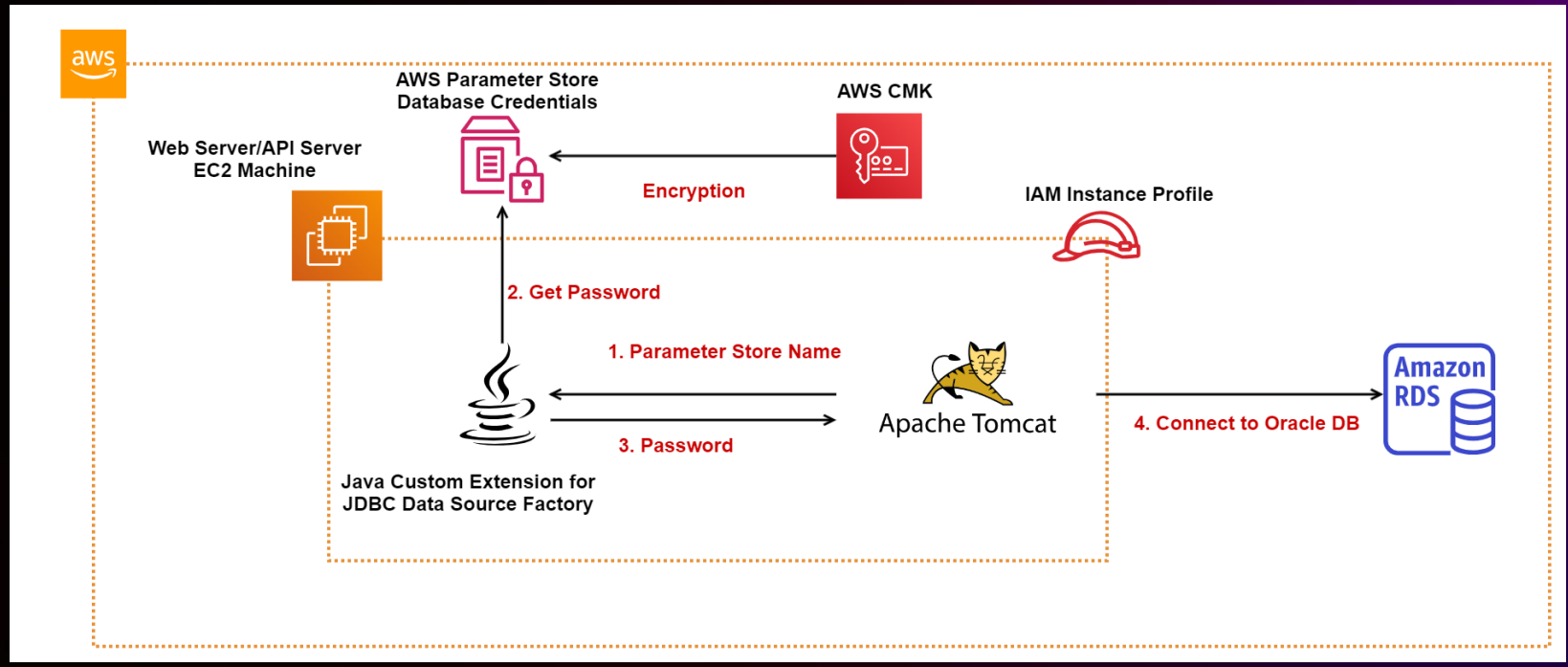


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Tomcat with AWS Parameter Store

- Dynamic password injection
- No plain text password

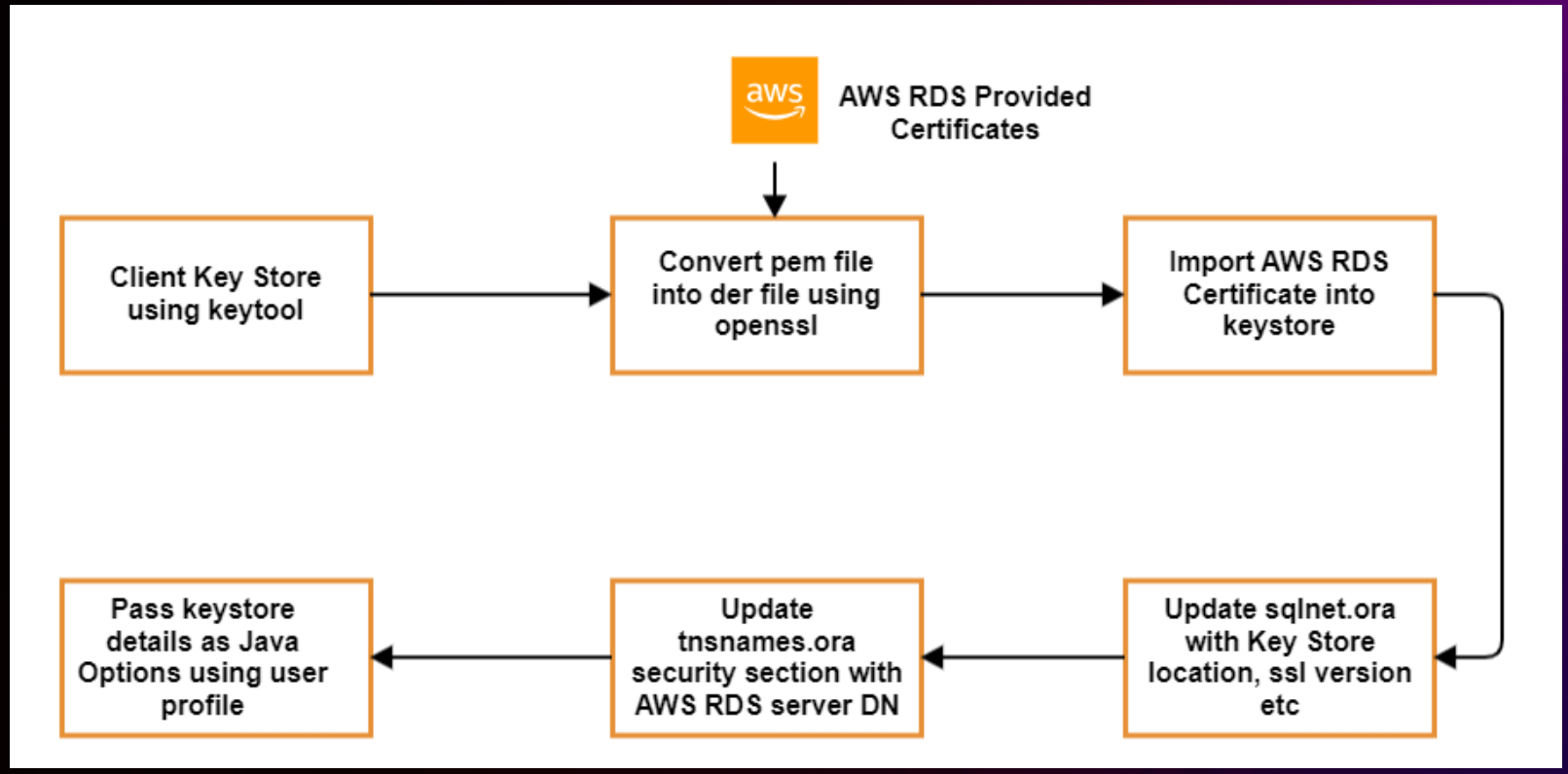


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Oracle app to Amazon RDS via TCPS

- In-transit encryption
- Use regional certificate
- Server DN match
- Keystore details via Java options



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Performance tuning - Amazon RDS

- Oracle AWR report
- Amazon RDS instance class and size
- Amazon RDS Performance Insights
- Database tuning
- SQL tuning/profile

INSTANCE CLASS	STORAGE	REDO LOG SIZE	TEMP TABLESPACE
db.r5.8xlarge	20 TB	160 GB	200 GB

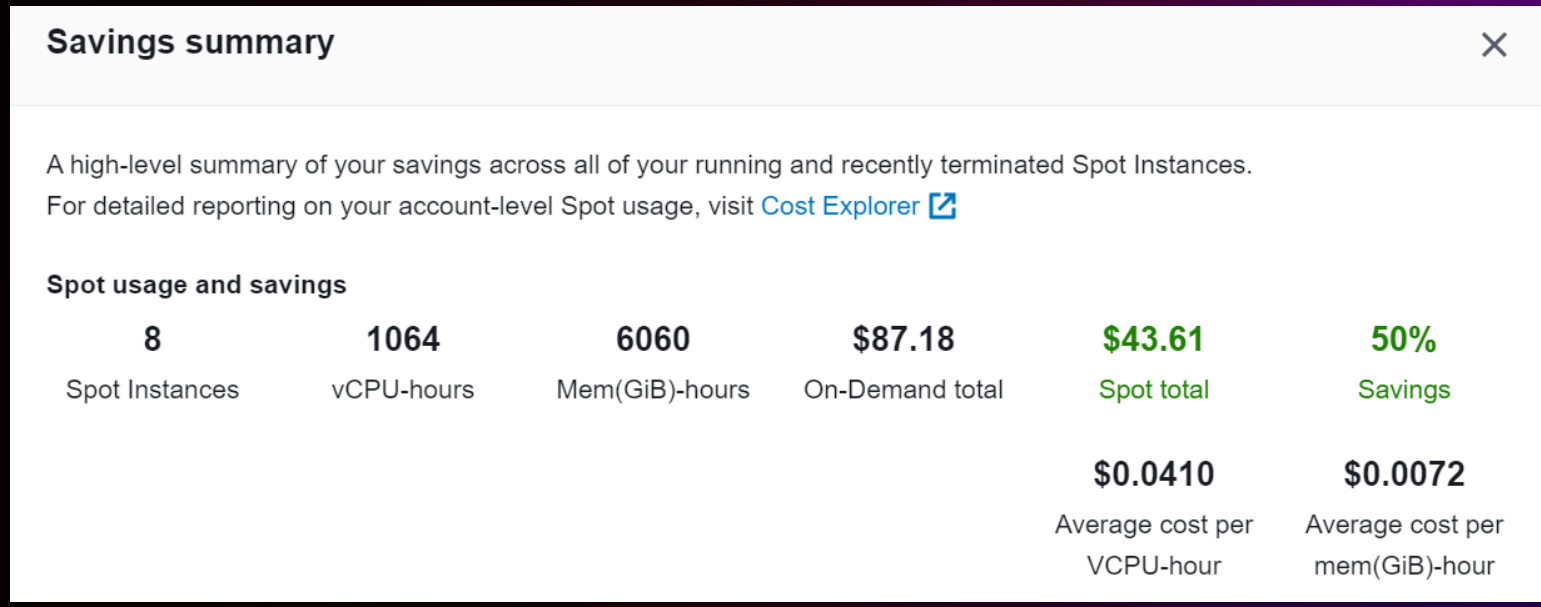
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

Spot Instances with Oracle apps

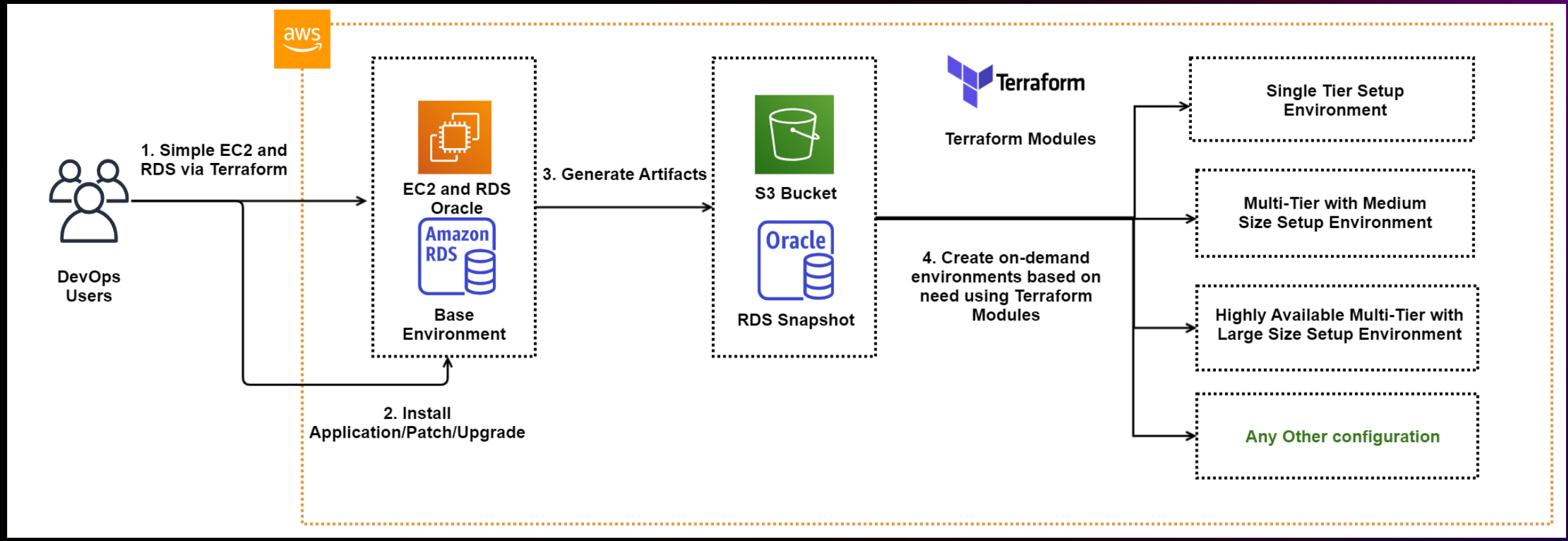
- Significant cost reduction
- ASG with Spot Instances
- Playground environments
  1. Spot request of type persistent
  2. Interruption behavior as stop



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Consistent environments

ORACLE FINANCIAL SERVICES ANALYTICS APPLICATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

national australia bank

New environment

ENVIRONMENT INPUTS

- Generic question and answer

test2 > {} inputs.json > {} ofsaa

```
{
  "version": "v2.2.0",
  "environment": "test2",
  "provisioner_role": "ofsaa-test2-ftp-build-role",
  "tfbackend": {},
  "cmk_iam": {
    "bastion_profile": true,
    "landingzone_efsaccess": true,
    "spotpowermgmt_role"   : null,
    "athena_glue_access": false

  },
  "bootstrapbucket": {},
  "ofsaa": {
    "enable_autoscaling": false,
    "enable_multitier": false,
    "enable_database_multi_az": false,
    "enable_database_readreplica":false,
    "enable_controlm": true,
    "enable_appd": true,
    "enable_docker": true,
    "enable_fluentd": true,
    "enable_landingzone": false,
    "enable_landingzone_archive": false,
    "enable_snapshotdb_tos3": false,
    "enable_spotinstance": ["app"],
    "db_snapshot_name": "ftptest2v811-v2",
    "create_bastion": false,
    "attach_efs": false,
    "ebs_cmk_arn_ssmps": null,
    "rds_cmk_arn_ssmps": null,
    "ssm_cmk_arn_ssmps": null,
    "ssmps_software_bucket_name": null,
    "s3_cmk_arn_ssmps": null,
    "instance_profile": null,
    "power_mgt": "Stop{20:30,1-5}",
    "power_mgt_rds": "EXT",
    "environment_size_servers": "medium",
    "environment_size_database": "small"
  }
}
```

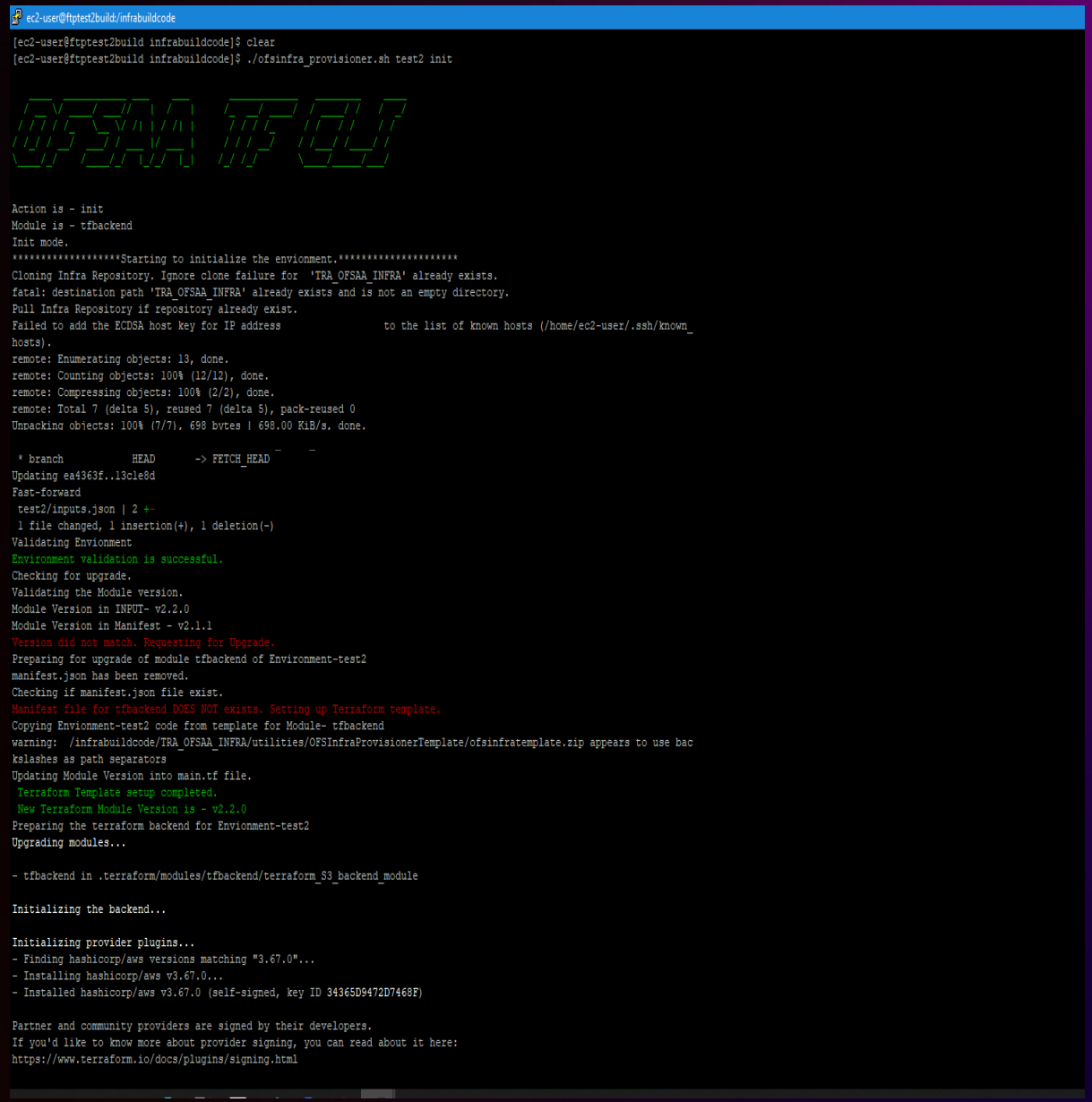
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New environment

INITIALIZE THE ENVIRONMENT

- Setup S3 bucket
- Terraform backend setup
- Secure state file



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New environment

GENERATE TERRAFORM CODE AND CREATE PLAN

ec2-user@ftptest2build:/infrabuildcode

```
[ec2-user@ftptest2build infrabuildcode]$  ./ofsinfra_provisioner.sh test2 config ofsaa


Action is - config
Module is - ofsaa
Config mode.
*******************Starting to configure the envionment.**********************
Environment validation is successful.
Checking for upgrade.
Validating the Module version.
Module Version in INPUT- v2.2.0
Module Version in Manifest - v2.1.1
Version did not match. Requesting for Upgrade.
Preparing for upgrade of module ofsaa of Environment-test2
manifest.json has been removed.
Manifest file for ofsaa DOES NOT exists. Setting up from Terraform template.
Copying Envionment-test2 code from template for Module- ofsaa
warning:  /infrabuildcode/TRA_OFSAA_INFRA/utilities/OFSInfraProvisionerTemplate/ofsinfratemplate.zip appears to use backslas
Updating Module Version into main.tf file.
 Terraform Template setup completed.
 New Terraform Module Version is - v2.2.0
Checking if Backend is moved to S3 bucket.
backend.tf exists. Backend is already moved to s3 bucket.
Upgrading modules...
Downloading                              :/ATNGIAPPRJ/TRA_OFSAA_TF_MODULES.git?ref=v2.2.0 for ftpofsaaa...
- ftpofsaaa in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module
- ftpofsaaa.athenafiles in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/s3bucket
- ftpofsaaa.bastion_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/bastionserver
- ftpofsaaa.bastion_keypair in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/keypair
- ftpofsaaa.common_tags_provider in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/commontagsprovider
- ftpofsaaa.global_parameter_store in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/globalparameterstore
- ftpofsaaa.ofs_keypair in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/keypair
- ftpofsaaa.ofsaa_api_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/ofsapiserver
- ftpofsaaa.ofsaa_app_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/ofsappserver
- ftpofsaaa.ofsaa_db_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/ofsdatabase
- ftpofsaaa.ofsaa_web_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/ofswebserver
- ftpofsaaa.ofsdb_archive_instance in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/ofsdatabase
- ftpofsaaa.s3archiveinboundlandingzone in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/s3bucket
- ftpofsaaa.s3archiveoutboundlandingzone in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/s3bucket
- ftpofsaaa.s3snapshotdbarchive in .terraform/modules/ftpofsaaa/ofsaa_ftp_infra_module/s3bucket

Initializing the backend...

Initializing provider plugins...
```

ec2-user@ftptest2build:/infrabuildcode

```
      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = true
          + device_name           = (known after apply)
          + encrypted             = true
          + iops                  = (known after apply)
          + kms_key_id            =
          + tags                  = {
              + "AppCategory"   = "C"
              + "ApplicationID"
              + "CostCentre"    = "v_ftpaws"
              + "CreatedBy"     = "terraform"
              + "Environment"   = "test2"
              + "InstanceID"    = "Placeholder"
              + "Name"          = "Placeholder"
              + "SupportGroup"  = "Risk Support-RTT Treasury Apps"
            }
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = 20
          + volume_type           = "gp3"
        }
    }

  # module.ftpofsaaa.module.ofsaa_app_instance.null_resource.spot_app_instance_service[0] must be replaced
+/- resource "null_resource" "spot_app_instance_service" {
      ~ id       = "5577006791947779410" -> (known after apply)
      ~ triggers = { # forces replacement
          ~ "instance_id"    =
          ~ "private_ip"     =
          ~ "spotrequest_id" =
        }
    }

Plan: 11 to add, 1 to change, 10 to destroy.

------------------------------------------------------------------------

This plan was saved to: out.terraform

To perform exactly these actions, run the following command to apply:
    terraform apply "out.terraform"

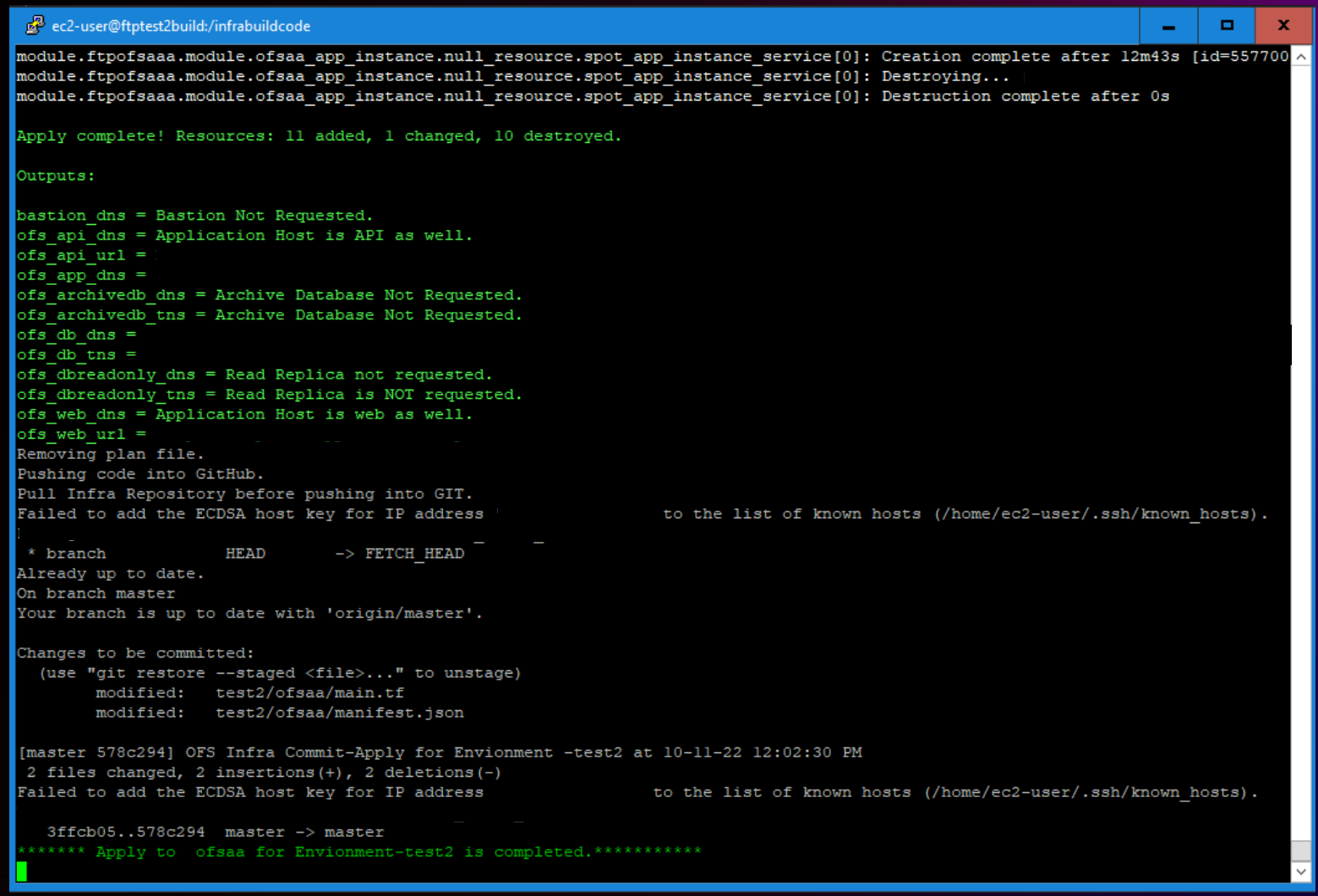
Please review plan output(terraform show out.terraform) and run apply ofsaa
******* Config to ofsaa for Envionment-test2 is completed.***********
[ec2-user@ftptest2build infrabuildcode]$
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New Environment

APPLY THE PLAN TO CREATE AWS RESOURCES

On prem	AWS
1-2 months	60 min

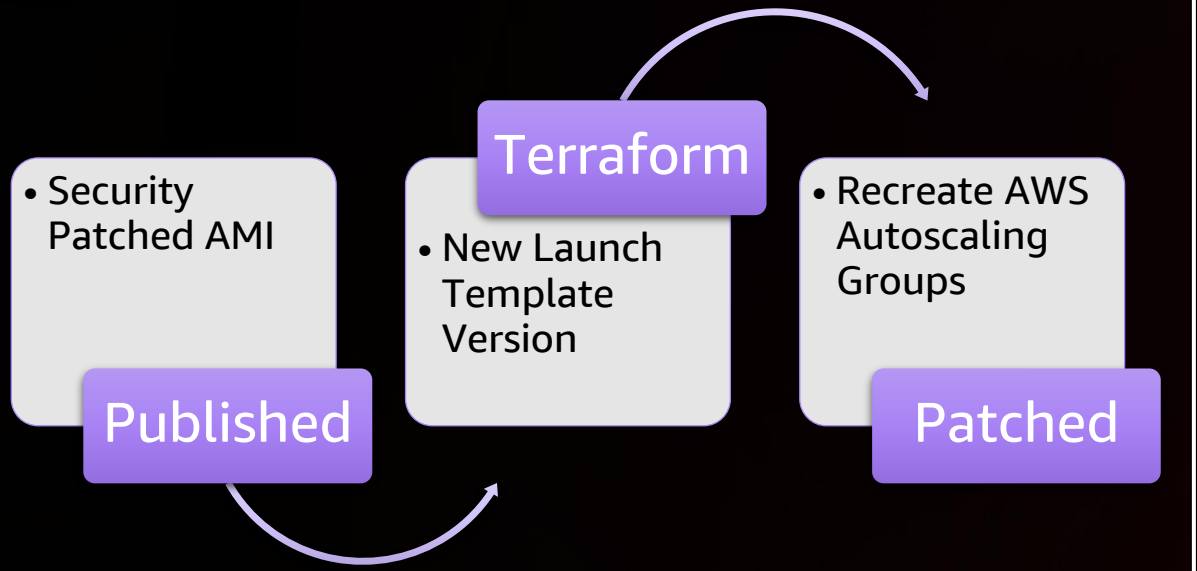


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

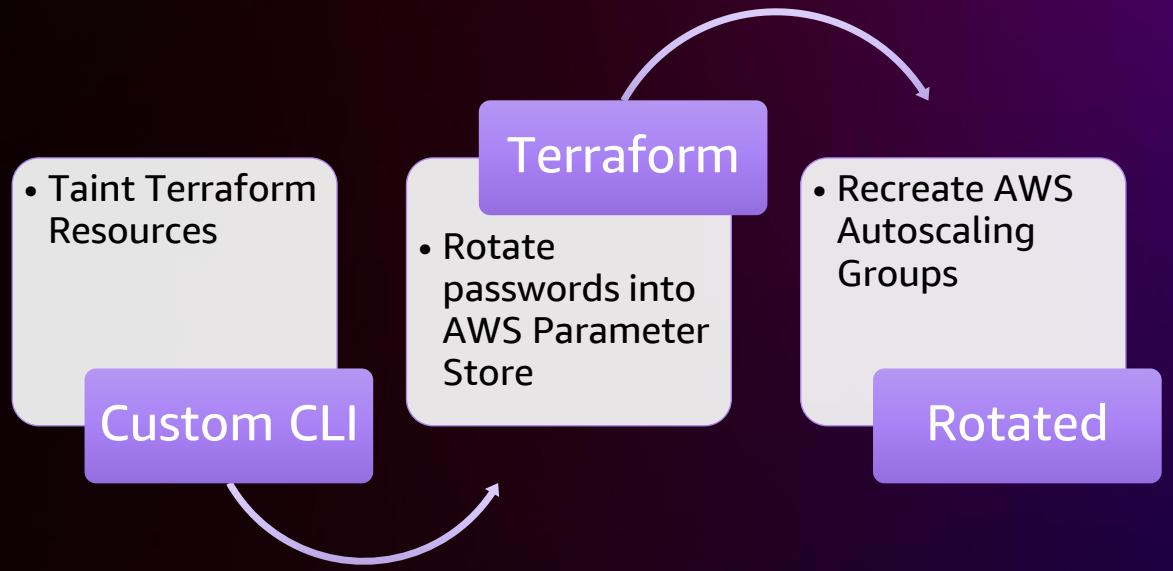
Maintaining environments

CUSTOM CLI TO PATCH AND ROTATE PASSWORDS

Automated patching



Automated password rotation



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Results achieved

- Regulatory compliance
- Agility in delivering business value
- SLA improved by 30%
- On-demand environments
- Highly secured infrastructure
- Significant cost reduction using Spot Instances

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What’s next?

AWS RDS CUSTOM FOR ORACLE E-BUSINESS SUITE

Amazon RDS Custom is a managed database service for legacy, custom, and packaged applications that require access to the underlying operating system and database environment

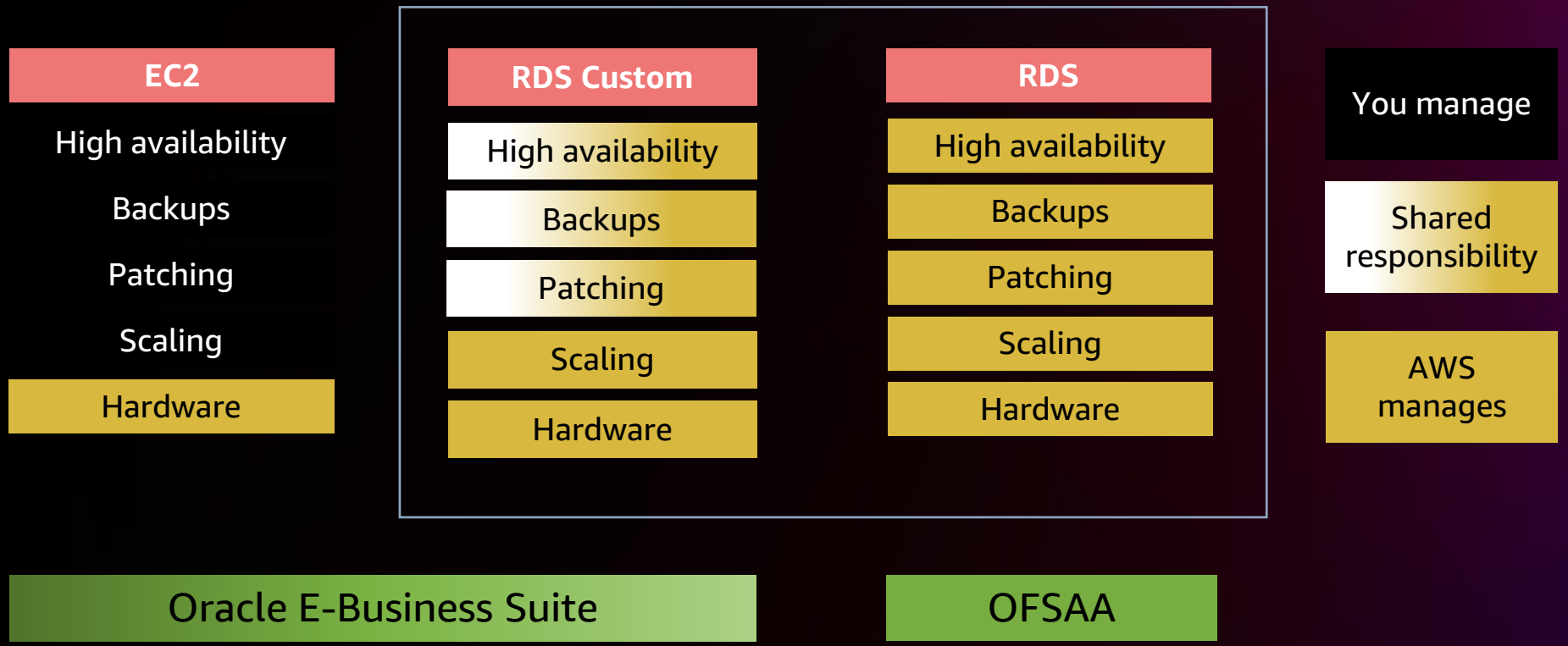


aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Database choice

MANAGED CLOUD DATABASE SERVICE



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Database choice

MANAGED CLOUD DATABASE SERVICE

EC2

High availability

Backups

Patching

Scaling

Hardware

AWS Architecture Blog

Setup a high availability design for Oracle Data Guard (Fast-Start Failover) using Amazon Route 53

by Viqash Adwani | on 09 SEP 2022 | in Amazon CloudWatch, Amazon EC2, Amazon Route 53, Architecture | Permalink | Comments | Share

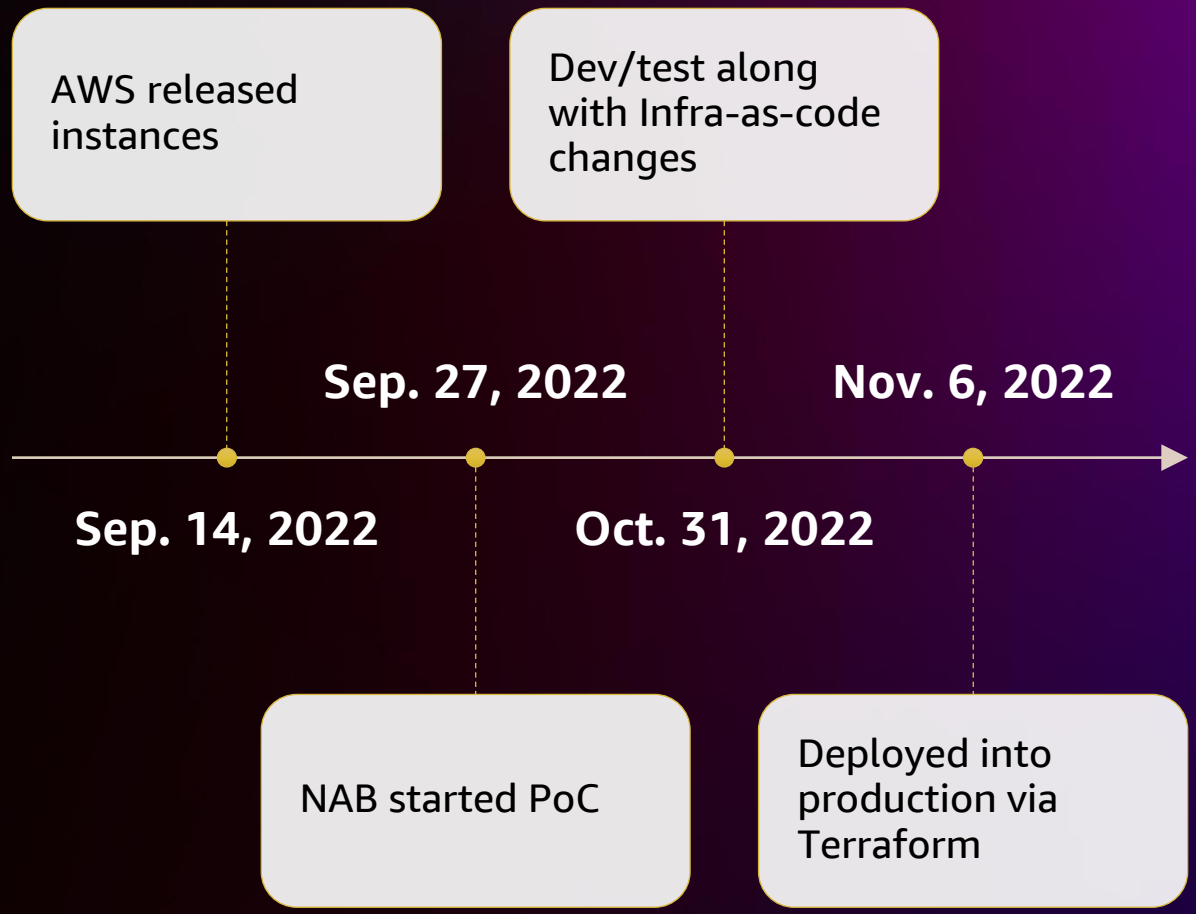
Many customers use Oracle Database deployed on Amazon Elastic Compute Cloud (Amazon EC2) to run their Oracle E-Business Suite applications. They rely on Oracle Data Guard for high availability databases, with a standby database running in a different availability zone. Oracle Data Guard can switch a standby database to the primary role in case a production database becomes unavailable due to planned/unplanned outage

https://aws.amazon.com/blogs/architecture/setup-a-high-availability-design-for-oracle-data-guard-fast-start-failover-using-amazon-route-53/

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What's next?

- Amazon RDS Database Smart Flash Cache with M5d and R5d instances
- Within 6 weeks, NAB was able to
  1. Upgrade Amazon RDS to latest minor version
  2. Use new instance class
  3. With minimal production database downtime (30-40 min)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

RDS Database Smart Flash Cache

Foreground Wait Events

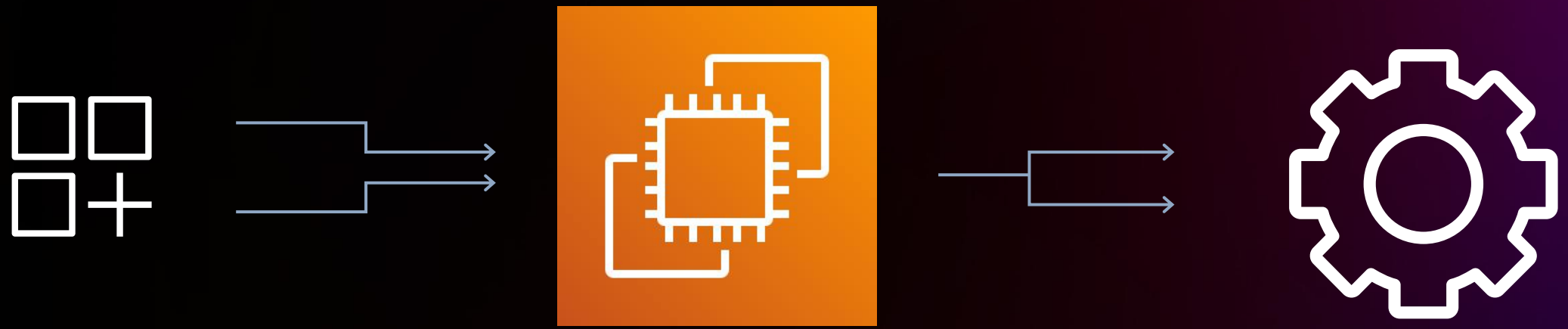
- s - second, ms - millisecond, us - microsecond, ns - nanosecond
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait	Waits /txn	% DB time
db file sequential read	4,476,624		6,525	1.46ms	205.33	52.87
direct path read	446,139		2,363	5.30ms	20.46	19.15
db flash cache multiblock physical read	553,438		381	689.16us	25.38	3.09
db flash cache single block physical read	555,202		66	118.97us	25.47	0.54
Failed Logon Delay	63	100	63	1000.07ms	0.00	0.51
db file scattered read	47,856		61	1.27ms	2.20	0.49

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What’s next?

- Multiple APIs for bank and partners based on Oracle Financial Service APIs



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Joshua Samuel
LinkedIn: @joshua-samuel

Ankit Singh
LinkedIn: @ankitdatascience

Robert Hanekom
LinkedIn: @roberthanekom



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.