
AWS
re:Invent

EUC304

Scaling and optimizing AppStream 2.0

Navi Magee
Senior Solutions Architect
End User Computing
AWS

Gaurav Kapoor
Senior Manager End User Services
Johnson & Johnson

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Objectives

Learn how to manage instance types

Learn how to scale and manage the AppStream 2.0 Fleet

Johnson & Johnson presentation and demo

Key takeaways

Conclusion



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

amazon AppStream 2.0

Manage instance types

Instance Family Types	Recommended Workloads
General Purpose	
Compute Optimized	
Memory Optimized	
Graphics Pro	
Graphics Design	SOLIDWORKS
Graphics G4	

“Using Amazon AppStream 2.0, our prospects can start a SOLIDWORKS trial in seconds instead of hours, without having to download, install, or configure anything on their computer.”

Todd Watts
Director, R&D, Web Applications, and Cloud Services
SOLIDWORKS

“AppStream 2.0 enables us to instantly deliver MATLAB to our workshop participants. Before using AppStream 2.0, customers had to manage unique software installations on different computers for every workshop. It could take up to 3 days to get everything ready. Now customers don't have to install or configure anything for a workshop. They simply open a browser on their computer and instantly get started. With AppStream 2.0, we can scale on-demand for any number of customers. AppStream 2.0 reduced the time it took to deploy our workshops by 75%.”



Roy Lurie

VP of Engineering, MATLAB Products

Running mode management

Always-on fleet

Instant connection

Uses running instance fleet

On-demand fleet

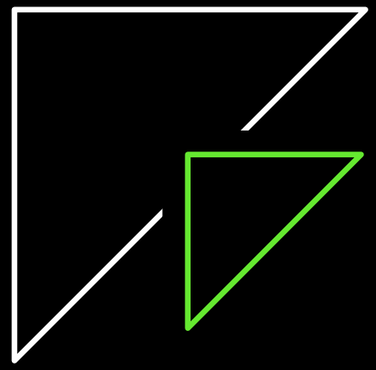
1–2-minute launch time

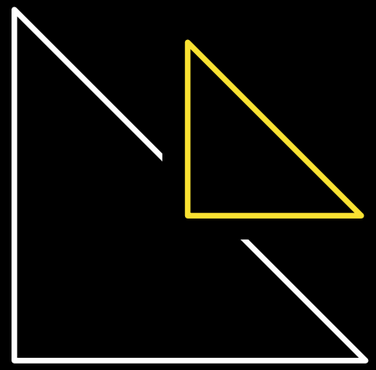
Uses stopped instance fleet

Which mode is right for me?

Scale and manage the AppStream 2.0 Fleet

Scale-out policies

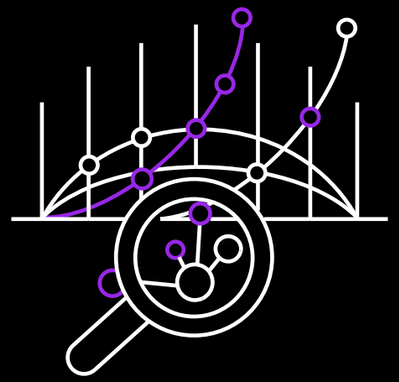




Scale-in policies

Scheduled policies

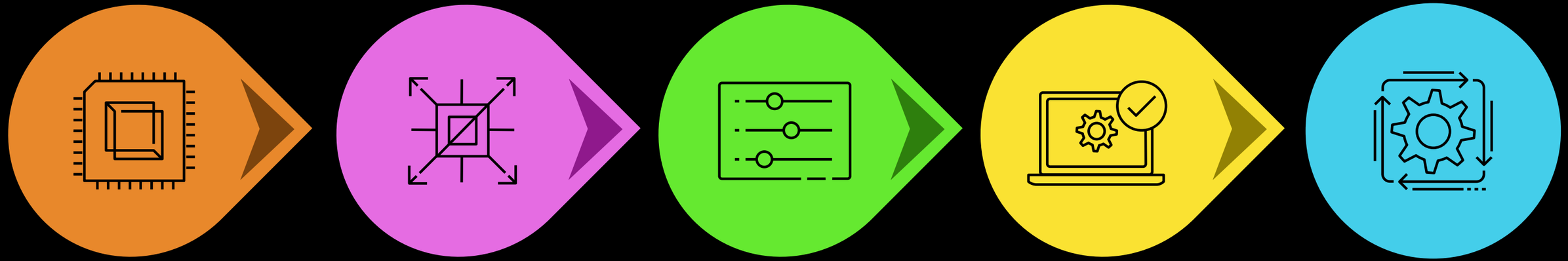




Target tracking policies

Optimize the AppStream 2.0 Fleet

5 STEPS TO OPTIMIZATION



Right-size the instance type

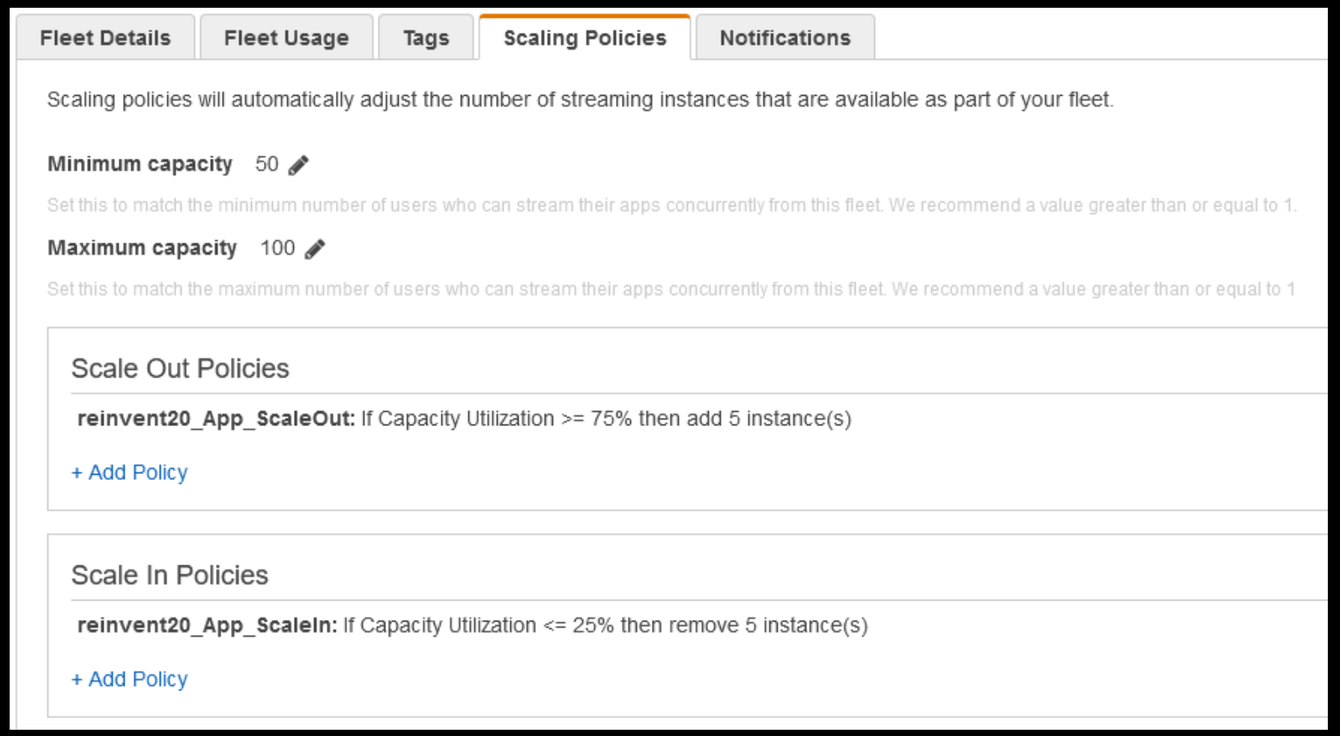
Adjust the fleet scaling size

Adjust the fleet scaling interval

Verify the running mode

Continuous monitoring

USE CASE



Requirements:

Deploy a set of applications for 100 users with 50% concurrency.

The users' work is time-sensitive and any delay accessing the applications would be costly for the company.

Fleet type:

Always-on


Johnson & Johnson

Johnson & Johnson

The largest and most diversified healthcare company in the world

Our Purpose

We blend ♡, Science and Ingenuity to profoundly change the trajectory of health for humanity.

Our People

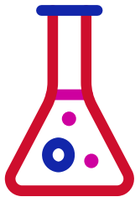


135,000 employees
who serve more than
1 billion patients each day

Strong Global Position in Three Key Segments



Consumer



Pharmaceuticals



Medical Devices

Use cases

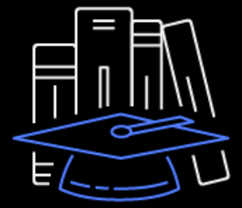
Today



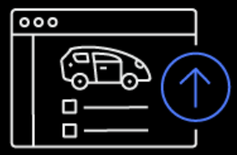
Secure BYOD Access



External Partner Access



Specialized Training

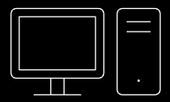


Offshore Engineering (CAD)

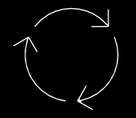
Emerging



Creo Parametric Suite (Desktop Experience)



Shared PC’s to thin-client (Desktop Experience)



DevOps, testing

UAT for J&J Software

Virtual learning experiences

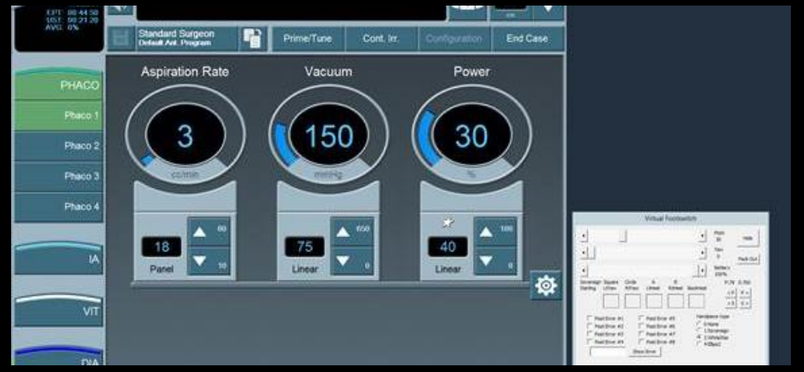
Software integrated solutions in Medical Devices

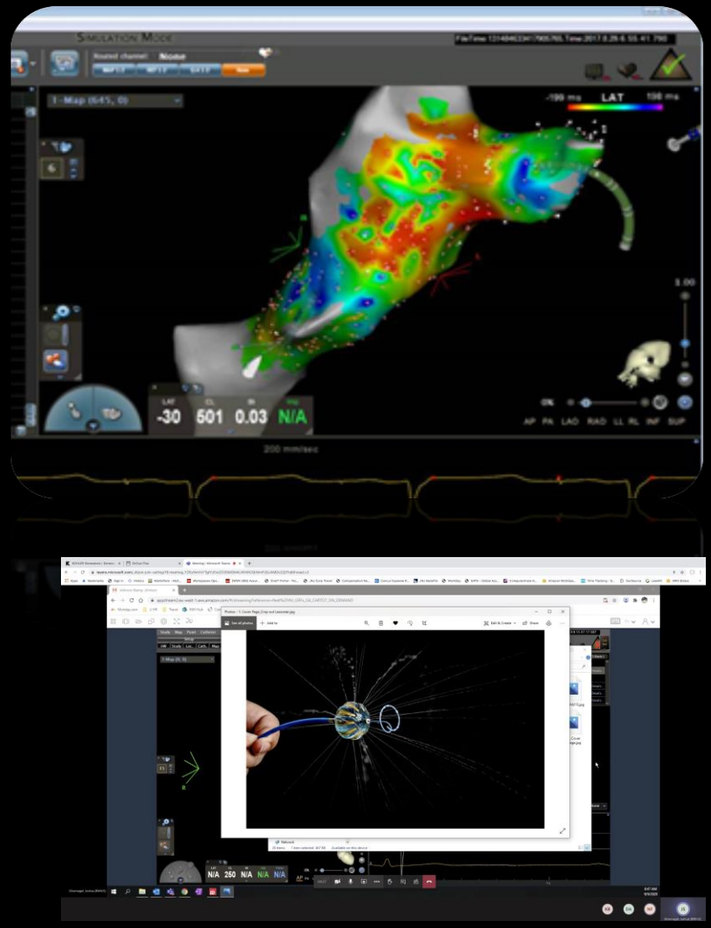
Need for travel reduced

Cloud Based Solutions remove the need for high end hardware

Digital health innovation

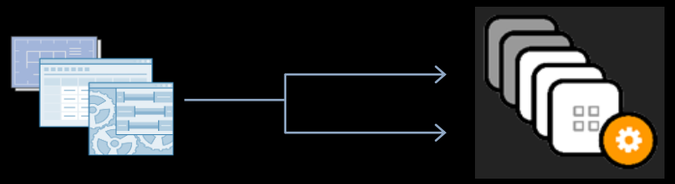
Perform live demos from any device





Our deployment

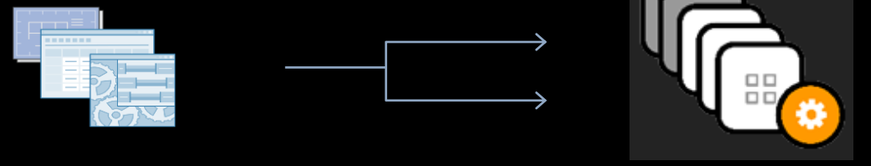
General Purpose Fleet



Common business apps and browsers

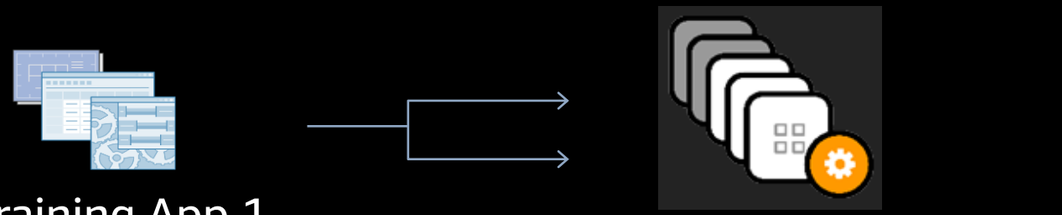
standard.medium

Dedicated Fleets



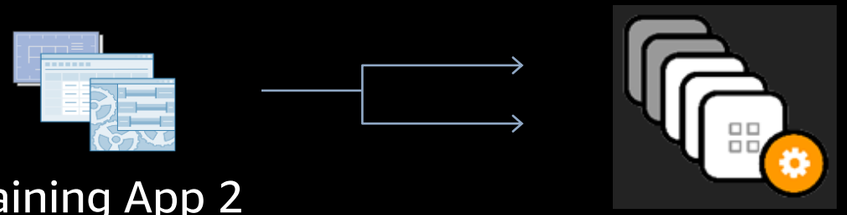
Complex Apps

compute.xlarge



Training App 1

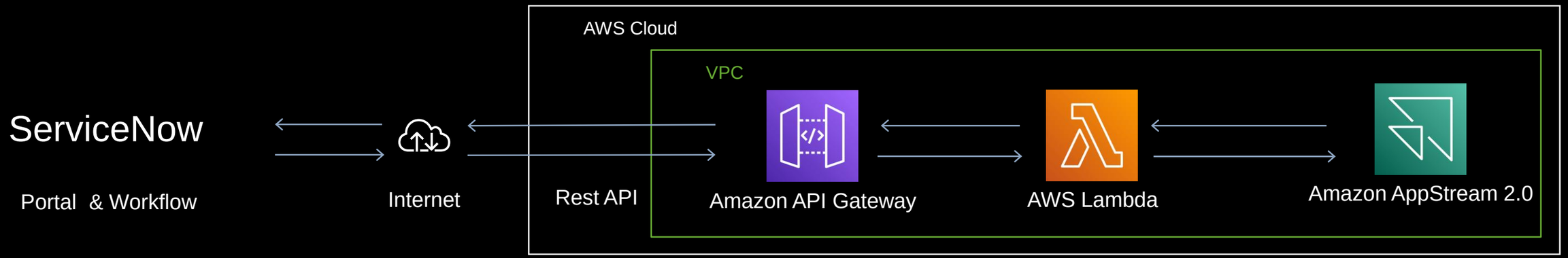
graphics-design.2xlarge



Training App 2

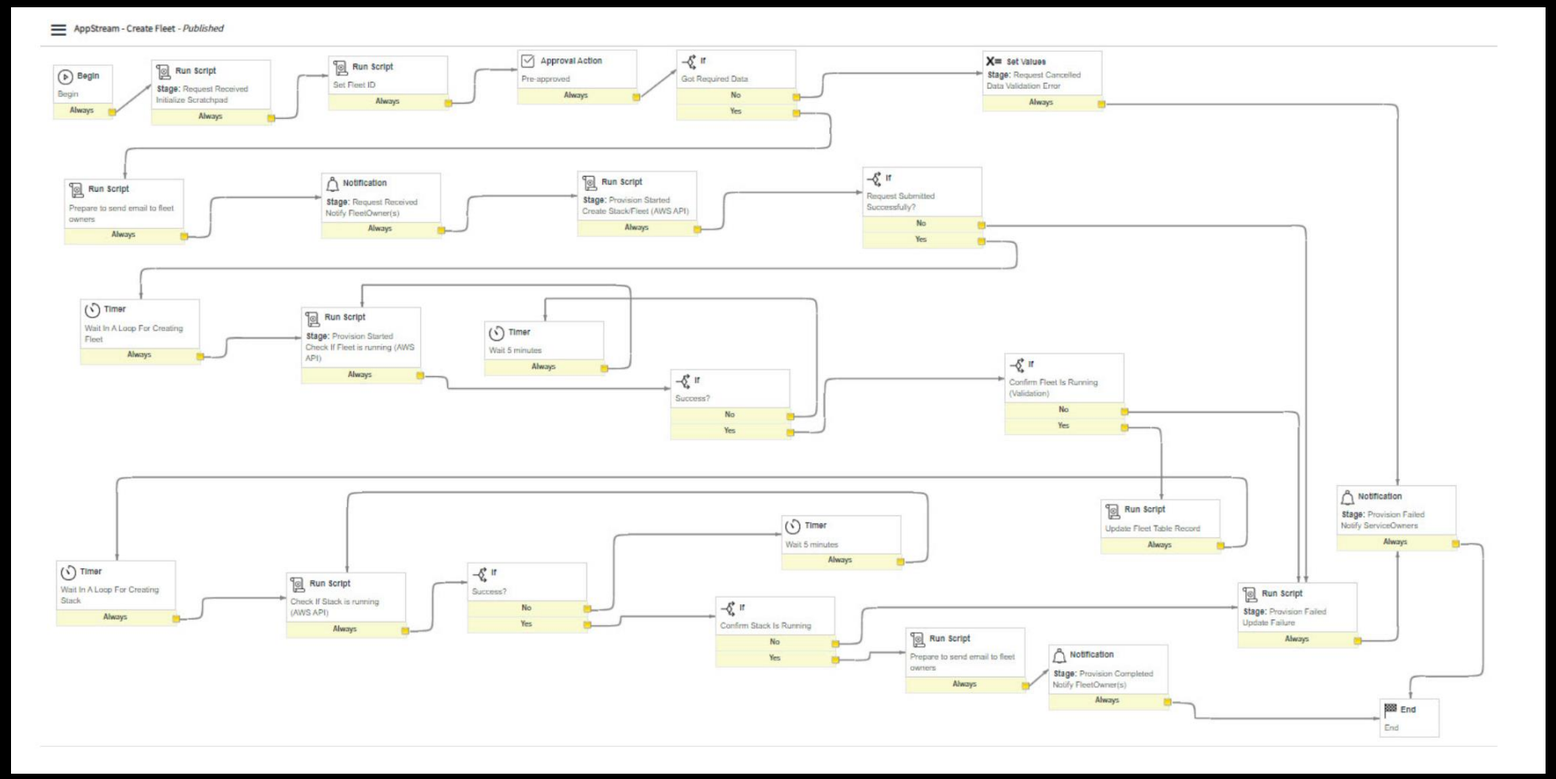
graphics-g4dn.xlarge

Self Service Architecture



Workflow

Sample Workflow and Code




```python
appstream = boto3.client('appstream')
def lambda_handler(event, context):
    print("Received event: " + json.dumps(event, indent=2))
    strFleetName = event['fleet_name']
    strImageName = event['image_name']
    strInstType = event['instance_type']
    strFleetType = event['fleet_type']
    strDisplayName = event['display_name']
    strDesiredInstances = event['desired_instances']
    strDisconnectTimeout = event['disconnect_timeout']
    strIdleDisconnectTimeout = event['idle_disconnect_timeout']
    strSubnetIds = event['subnetids']
    strSecurityGroups = event['securitygroups']
    try:
        response = appstream.create_fleet(
            Name=strFleetName,
            ImageName=strImageName,
            DisplayName=strDisplayName,
            FleetType=strFleetType,
            ComputeCapacity={'DesiredInstances': strDesiredInstances },
            VpcConfig={
                'SubnetIds':[strSubnetIds],
                'SecurityGroupIds':[strSecurityGroups]
            },
            DisconnectTimeoutInSeconds=strDisconnectTimeout,
            IdleDisconnectTimeoutInSeconds=strIdleDisconnectTimeout,
            InstanceType=strInstType
        )
        # create json response string
        responseStr = json.dumps(response, indent=4, sort_keys=True, default=str)
        return {
            'statusCode': 1,
            'message': "success",
            'body': responseStr
        }
    except Exception as ex:
        message = "Error creating Fleet"
        return {
            'statusCode': -1,
            'message': message,
            'body': json.dumps(ex, indent=4, sort_keys=True, default=str)
        }
```

myAppStream

Welcome to myAppStream!

As a first-time user, please read below to know about myAppStream service.

AppStream is a fully managed application streaming service that provides users with instant access to their desktop applications from anywhere. AppStream manages the AWS resources required to host and run your applications, scales automatically, and provides access to your users on demand.

Follow these steps to setup your applications in AppStream.

Step 1: Setup a profile with your project cost center and additional support contacts in Profile title.

Step 2: Request image builder to install and configure your applications in Image title.

Step 3: Once image is created, test your application for user experience by launching a QA fleet in Fleet title.

Step 4: Certify the image confirming your applications work in Fleet tile

Step 5: Create Production fleet with necessary settings for streaming your applications to your end users in Fleet tile.

Follow user guide, and FAQs for support questions.

Don't show me again

OK

© 2020 Johnson & Johnson, Inc. All rights reserved.

Takeaways



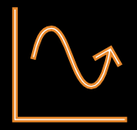
One size doesn’t fit all



Cost vs User Experience is important



Let the application owner drive



Scaling rules will vary for each use case



Virtual learning is part of the new normal

Recap



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

References

Scale your Amazon AppStream 2.0 Fleets

https://aws.amazon.com/blogs/desktop-and-application-streaming/scale-your-amazon-appstream-2-0-fleets/

Amazon AppStream 2.0 - Enterprise Application Onboarding Workshop

https://aws.amazon.com/getting-started/hands-on/app-onboarding/

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.