EU-01

Using DevOps methods to manage Amazon WorkSpaces and Amazon AppStream 2.0

Andrew Wood
Principal EUC Specialist SA
Amazon Web Services

The challenge for empowering the modern workforce

More **distributed** and **flexible** workforce

Need to improve **productivity** while reducing **cost**

While raising the bar on **security** and **reliability**

The right solution for your user personas

Amazon WorkSpaces

Cloud native persistent desktops

User gets their own desktop

Same instance every time the user logs in
User customizations persist between logins
Managed like a desktop or laptop
Deployed for each named user

Amazon WorkSpaces Web

Cloud native secure browser based access

User may only access non persistent browser

Secure access for users in browser based productivity environments
Automatically managed capacity & scaling, and updated
Enterprise policy and session controls on user interactions

Amazon AppStream 2.0 (desktop view)

Cloud native non-persistent desktops

User gets a new desktop every time

New instance each time the user logs in
Only admin-chosen customizations persist between logins
Managed like a kiosk
Deployed for expectation of multiple users

Amazon AppStream 2.0 (app view)

Cloud native apps

User sees just their applications every time

New instance each time the user logs in
Only admin-chosen customizations persist between logins
Managed like a kiosk
Deployed for expectation of multiple users

Cattle vs pets

Amazon AppStream 2.0

Cloud native non-persistent desktops

Amazon WorkSpaces

Cloud native persistent desktops

Why DevOps?

More **distributed** and **flexible** workforce

To scale resources for end users to work effectively

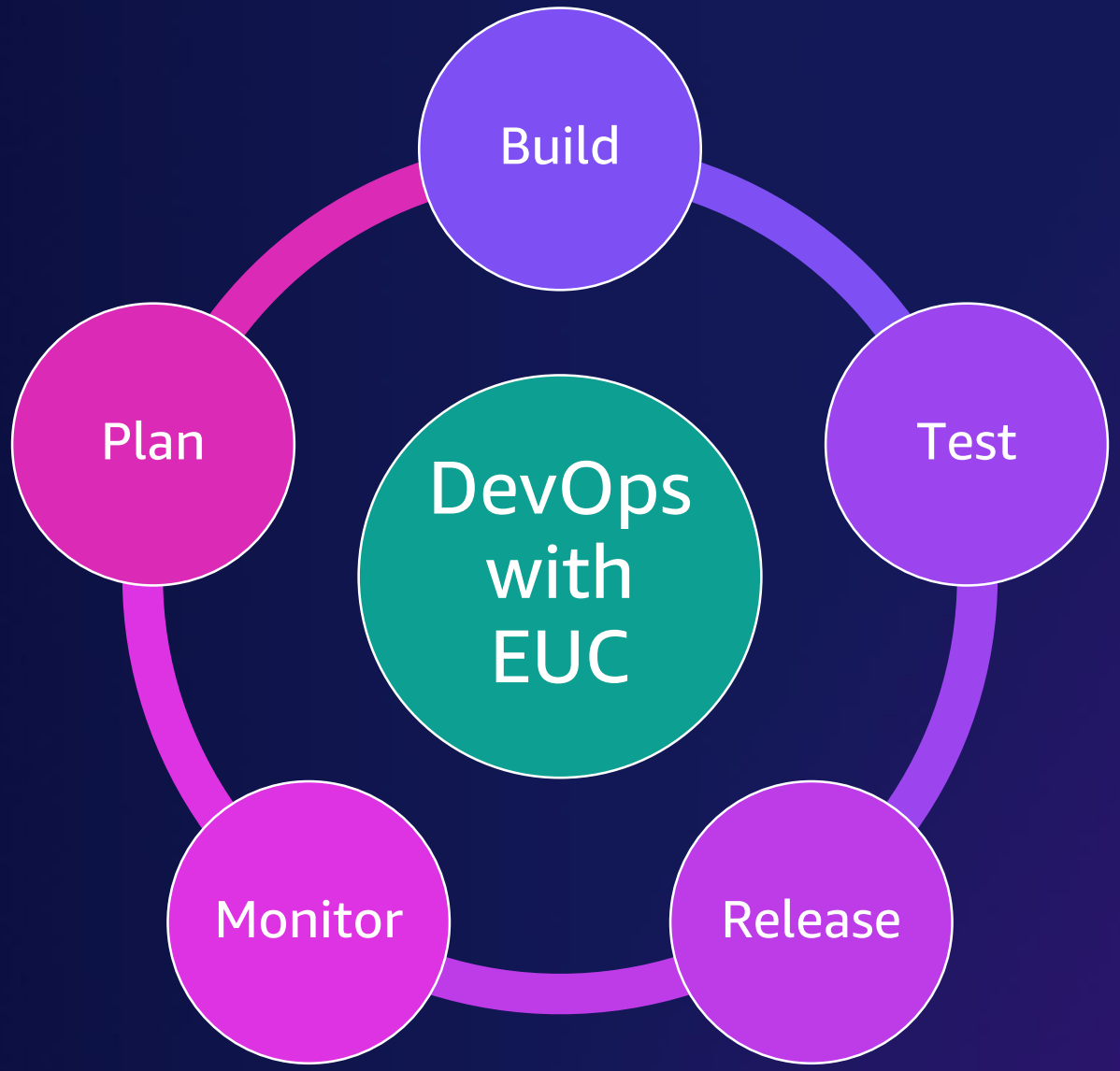
Need to improve **productivity** while reducing **cost**

Right sized resources with measurement to build a cost effective pay as you go model

While raising the bar on **security** and **reliability**

automate and reuse to reduce effort, optimize and measure reliability

Desktops and application delivery with DevOps



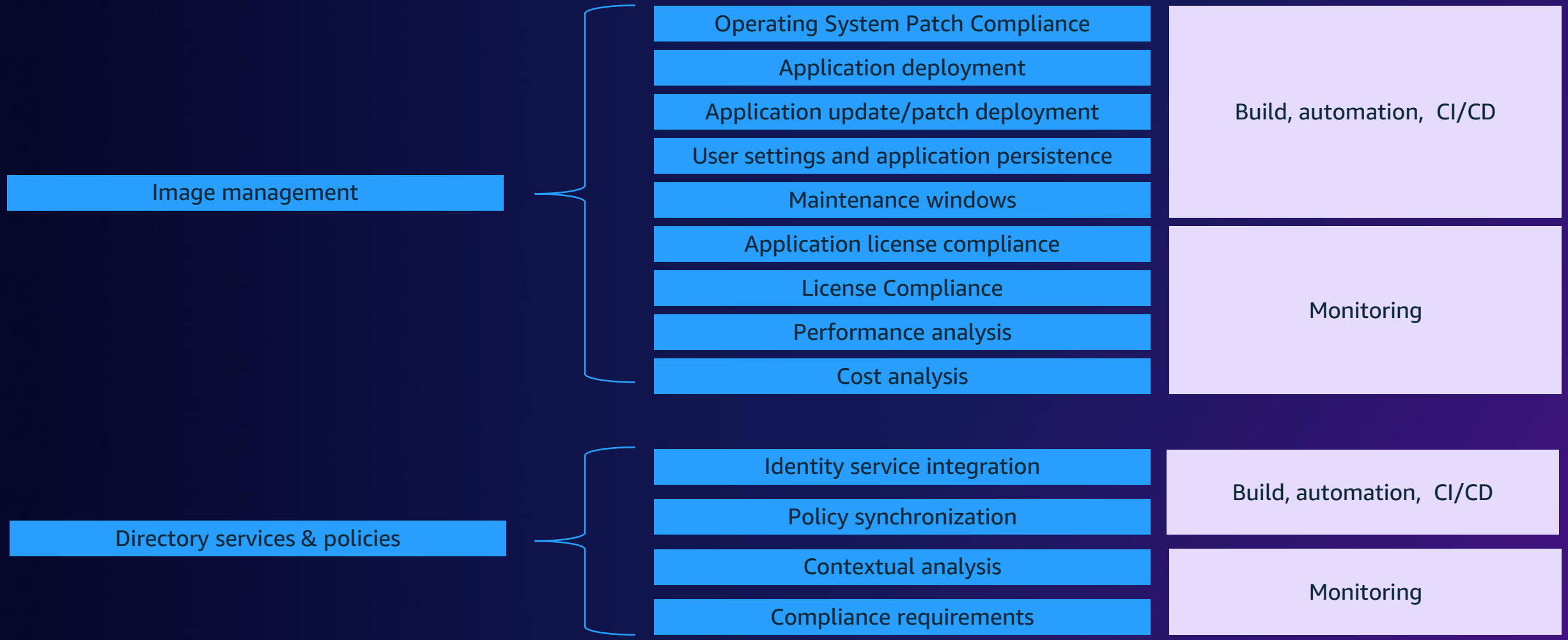
Responsibility for a fully managed service

Amazon – virtual applications and desktops as a service	Traditional cloud VDI	On-premises VDI
Image management	Image management	Image management
Directory services & policies	Directory services & policies	Directory services & policies
VDI control plane install & admin	VDI control plane install & admin	VDI control plane install & admin
Host admin	Host admin	Host admin
Storage admin	Storage admin	Storage admin
Load balancers install & admin	Load balancers install & admin	Load balancers install & admin
Hypervisor install & admin	Hypervisor install & admin	Hypervisor install & admin
Physical security	Physical security	Physical security
Power, HVAC	Power, HVAC	Power, HVAC
Rack and stack	Rack and stack	Rack and stack

Customer-managed

Service-managed

I’m managing…what now?



Build, test, release

Phases of automation

Manual

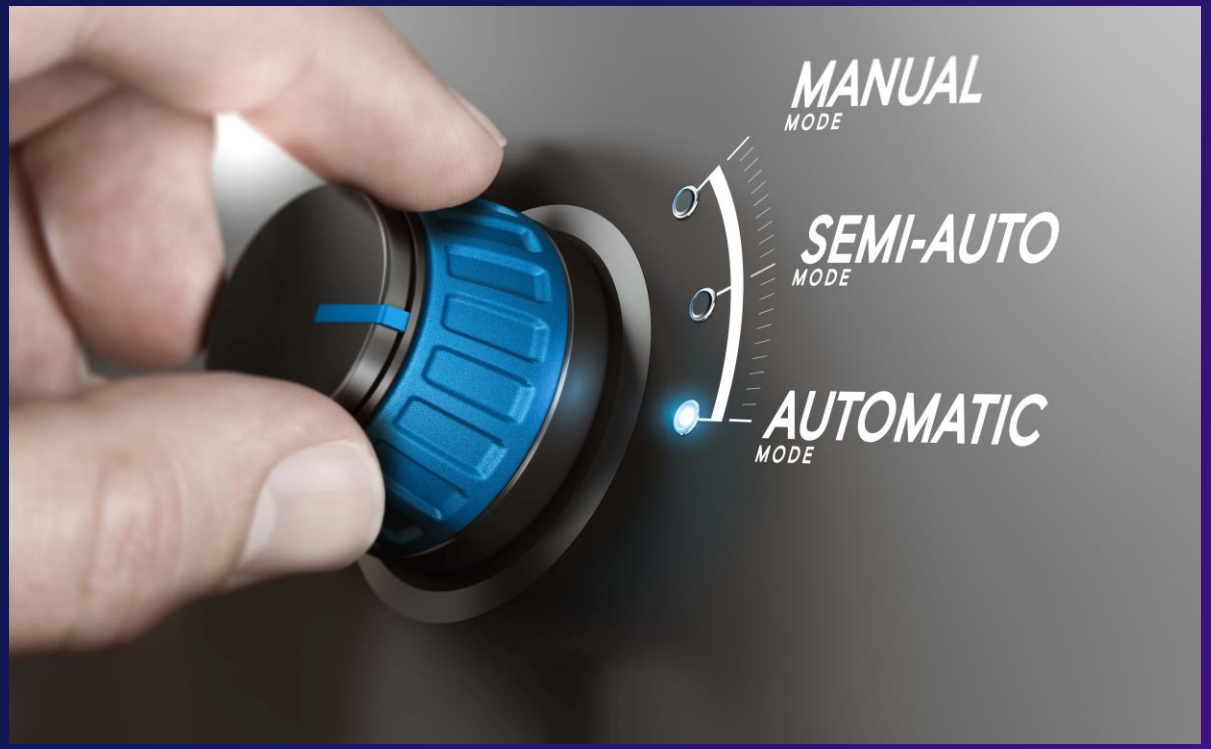
All of the components required are created in the console.

Partially Automated

The infrastructure components are programmatically created. The image creation portion is manual.

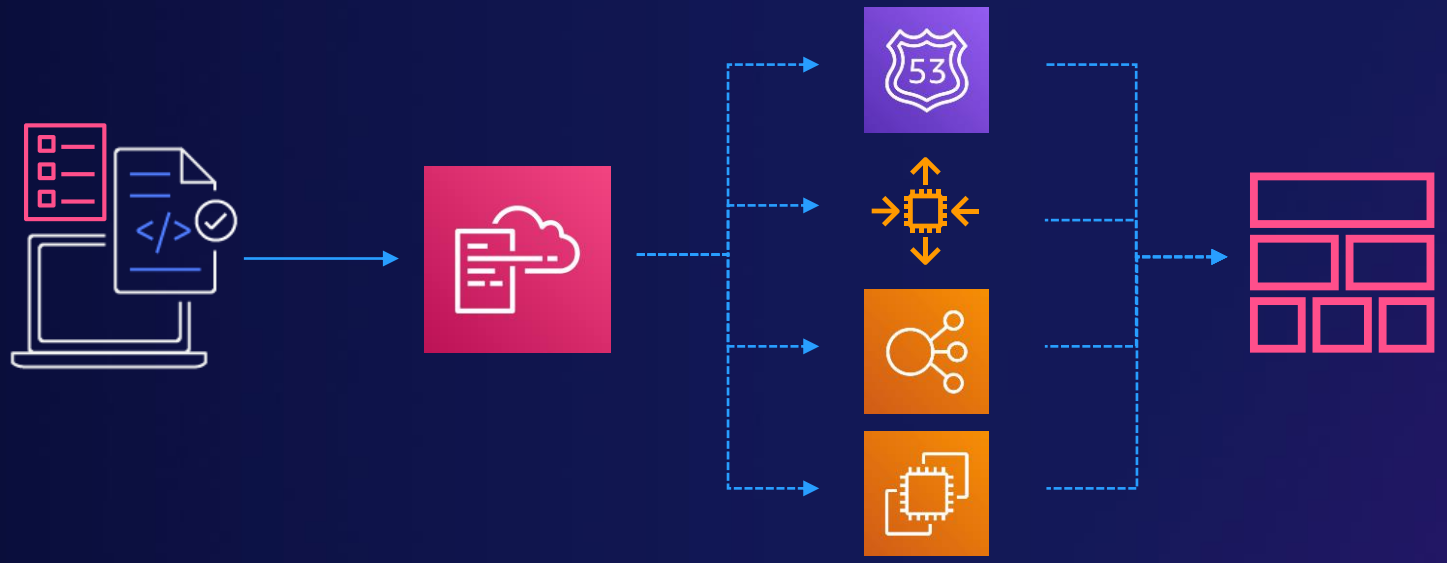
Full

No manual effort required to build and maintain the entire environment from end to end.



AWS CloudFormation

BUILD



Code in YAML or JSON directly or use sample templates

Upload local files or from an S3 bucket

Create stack using API via AWS CloudFormation

Stacks and resources are provisioned as a running environment

- JSON/YAML format template
- Presents template to AWS CloudFormation
- AWS CloudFormation translates it to an API request
- Forms a stack of resources

- FREE – you only pay for resources
- All regions
- APIs are called in parallel
- Manages **dependencies/relationships**

Cloud Formation with Amazon EUC

BUILD

- Amazon Directory Services

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-directoryservice-microsoftad.html

- Amazon WorkSpaces

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-workspaces-workspace.html

- Amazon AppStream 2.0

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-appstream-stack.html

```
AWSTEMPLATEFORMATVERSION: "2010-09-09"

DESCRIPTION: "THIS CLOUDFORMATION STACK CREATES
APPSTREAM RESOURCES WITHIN THE DEFAULT VPC."

RESOURCES:

  APPSTREAMFLEET:

    TYPE: "AWS::APPSTREAM::FLEET"

    PROPERTIES:

      NAME: "DEMOFLEET"

      DESCRIPTION: "THIS IS A DEMO FLEET THAT WAS
CREATED USING CLOUDFORMATION"

      DISPLAYNAME: "A DEMO FLEET CREATED IN
CLOUDFORMATION"

      IMAGENAME: "MYSAMPLEIMAGE"

      INSTANCETYPE: "STREAM.STANDARD.MEDIUM"

      FLEETTYPE: "ALWAYS_ON"

      COMPUTECAPACITY:

        DESIREDINSTANCES: 5

      VPCCONFIG:
```

The CI/CD disconnect

There is a common detachment between deploying infrastructure and automating the image creation

Build Pipeline

- VPCs, subnets, route tables
- IAM roles & policies
- Security groups
- AppStream fleets & stacks

- Install apps and settings
- Configure applications
- Create image

Image Creation

Release Pipeline

- Deploy images to QA environment
- Acceptance testing
- Approvals
- Release image to production

AWS CloudFormation

HashiCorp Terraform

CHEF

Red Hat Ansible

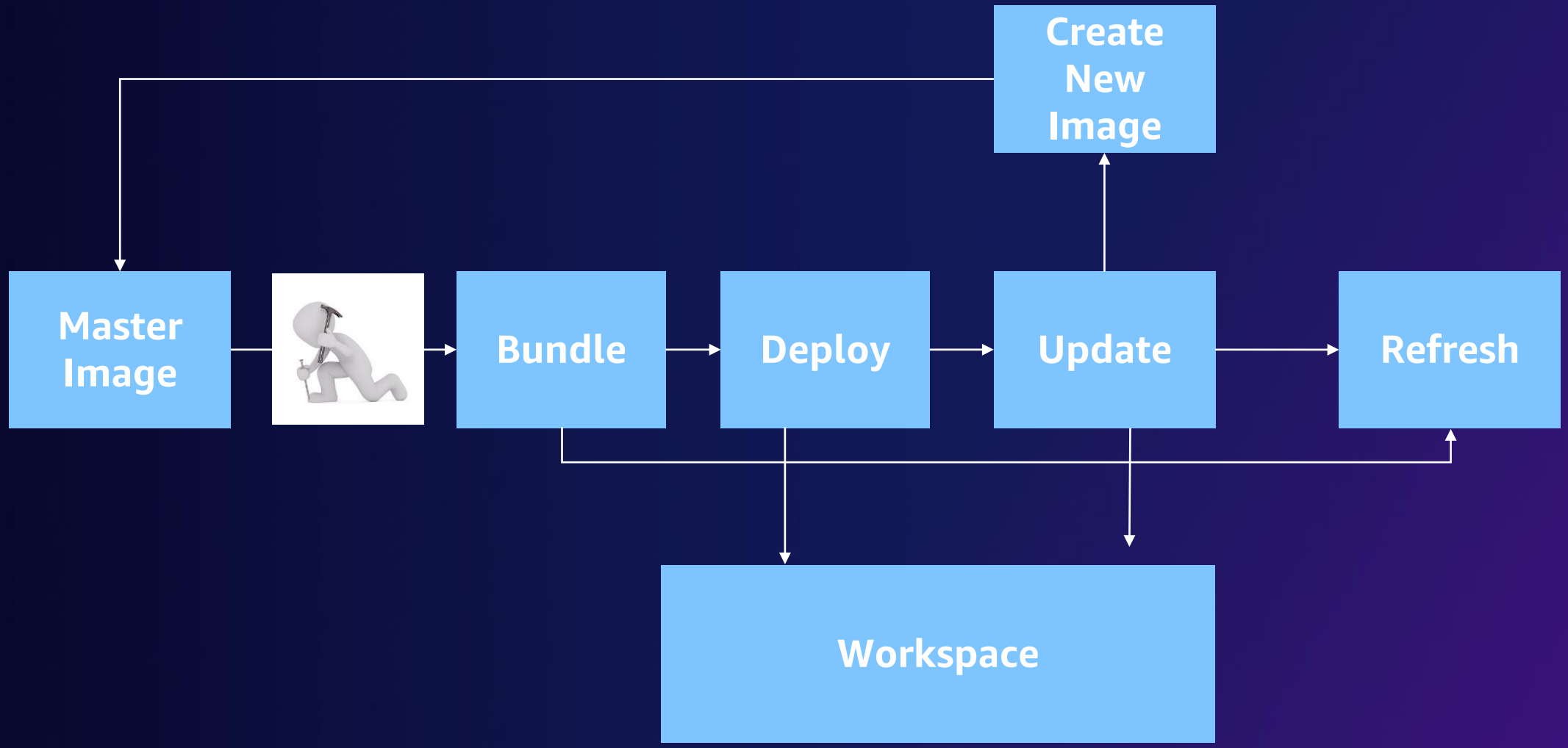
puppet

Jenkins

SALTSTACK

WorkSpaces Image Maintenance - Workflow

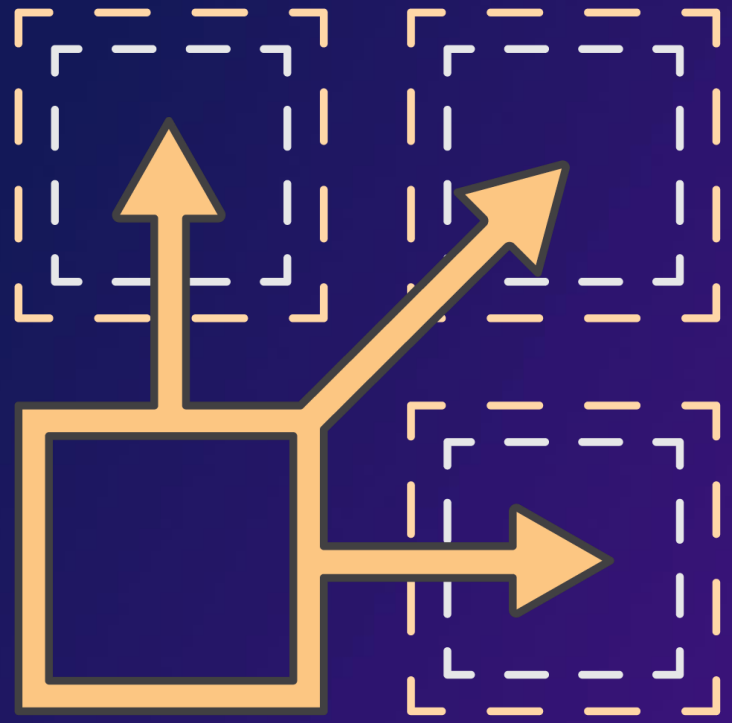
BUILD



The best way to deploy applications?

BUILD

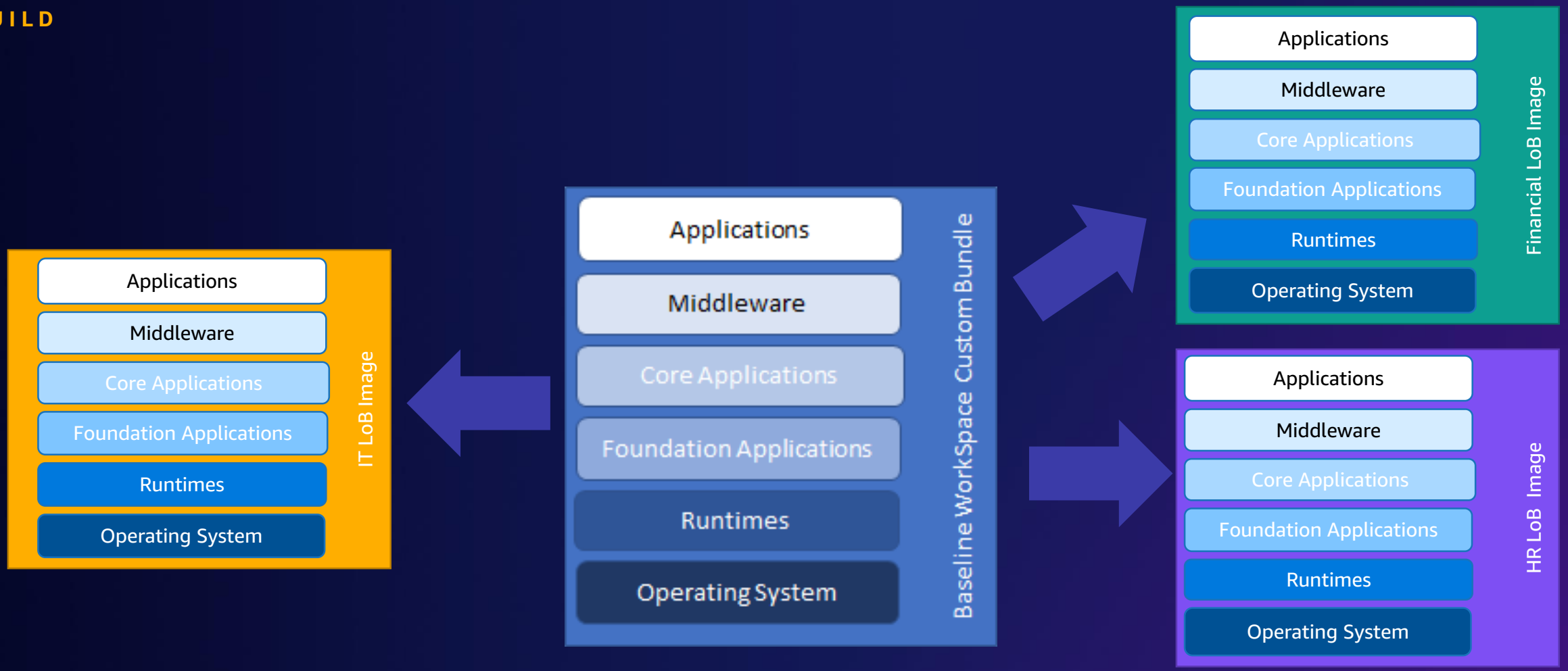
- **Separate the applications from the operating system?**



- No persistent changes to base bundle
- Deliver on demand
  - curated repository
  - Full self-determination
- Rapidly re-purpose
- Simplify application migration
- Portability and recovery for Business Continuity, and Disaster Recovery

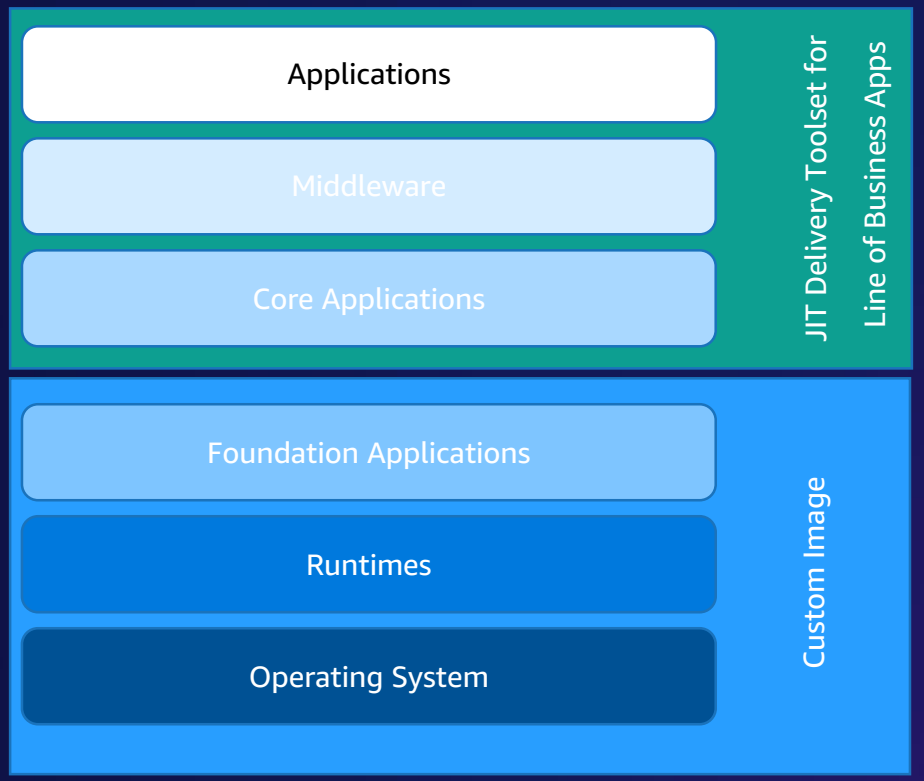
Application management

BUILD



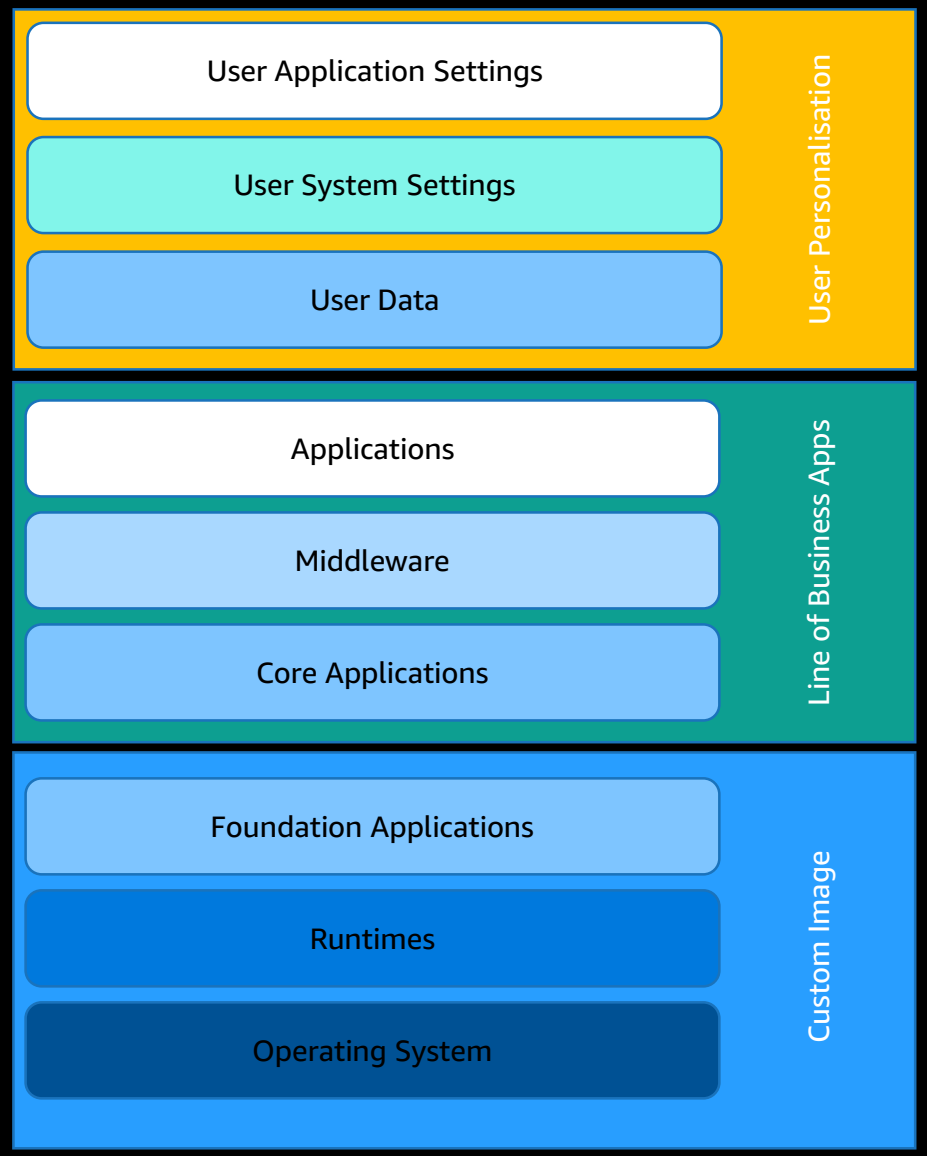
Application management

BUILD



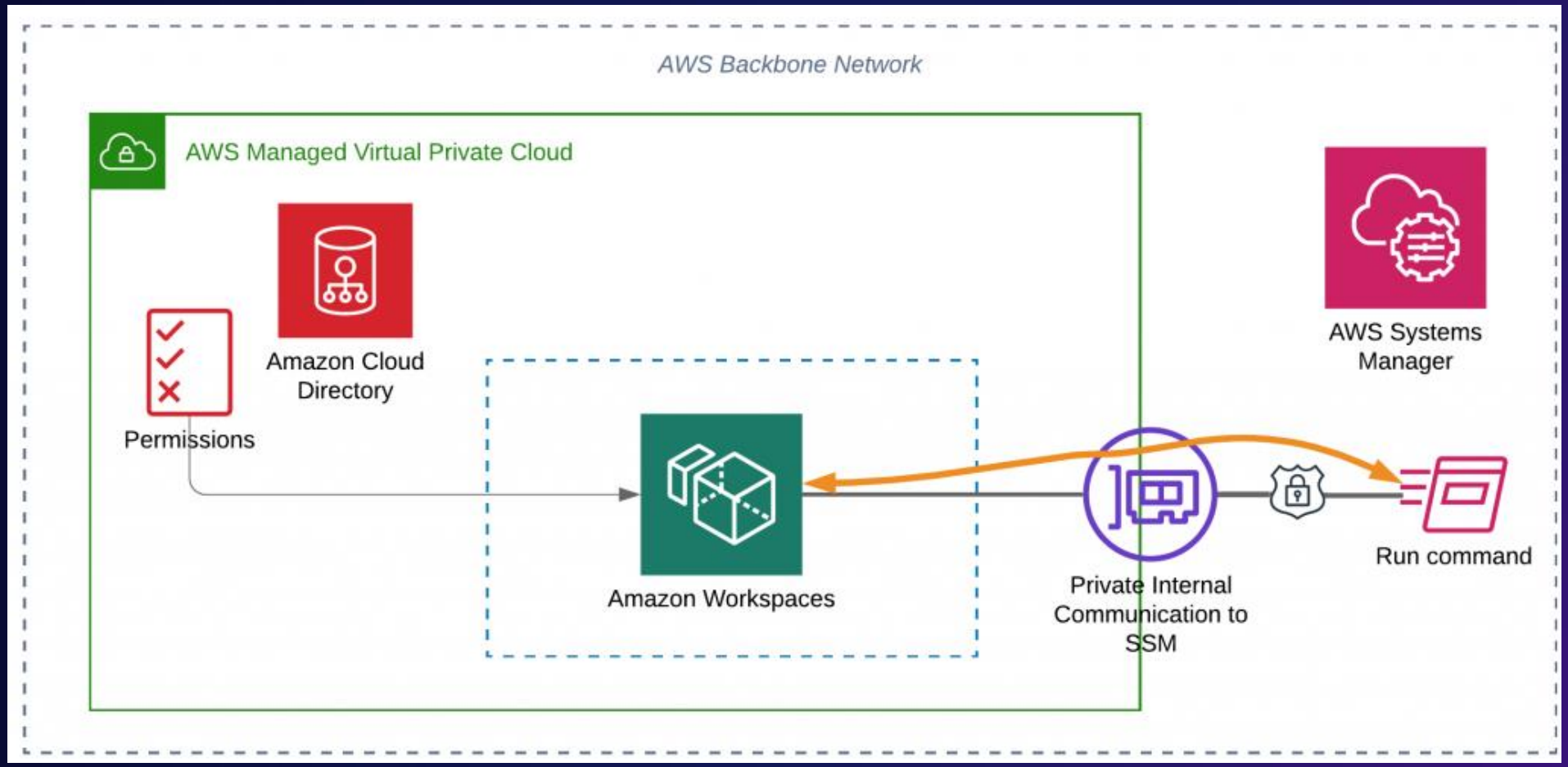
Profiles

BUILD



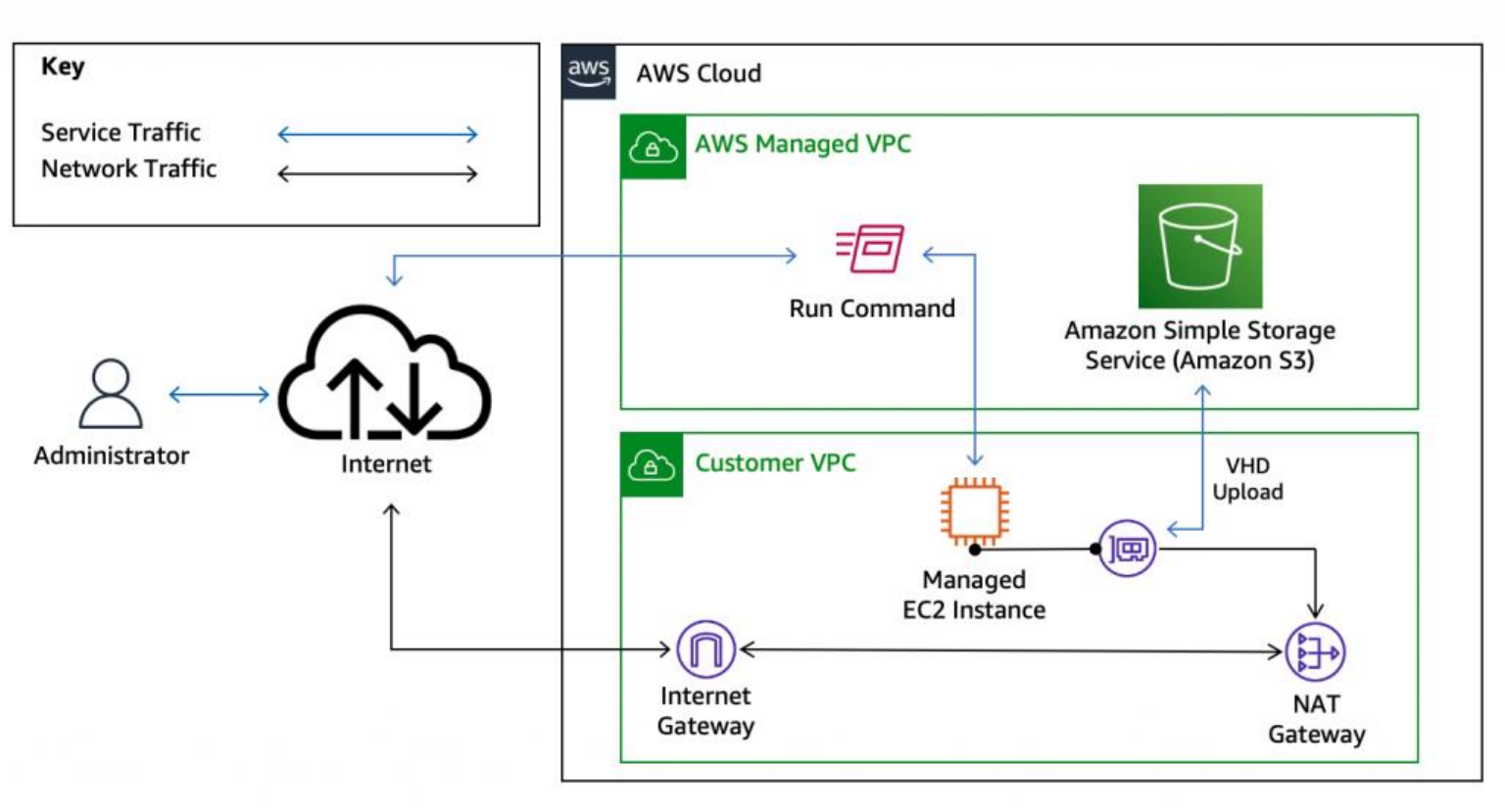
WorkSpaces with AWS Systems Manager

WORKSPACES



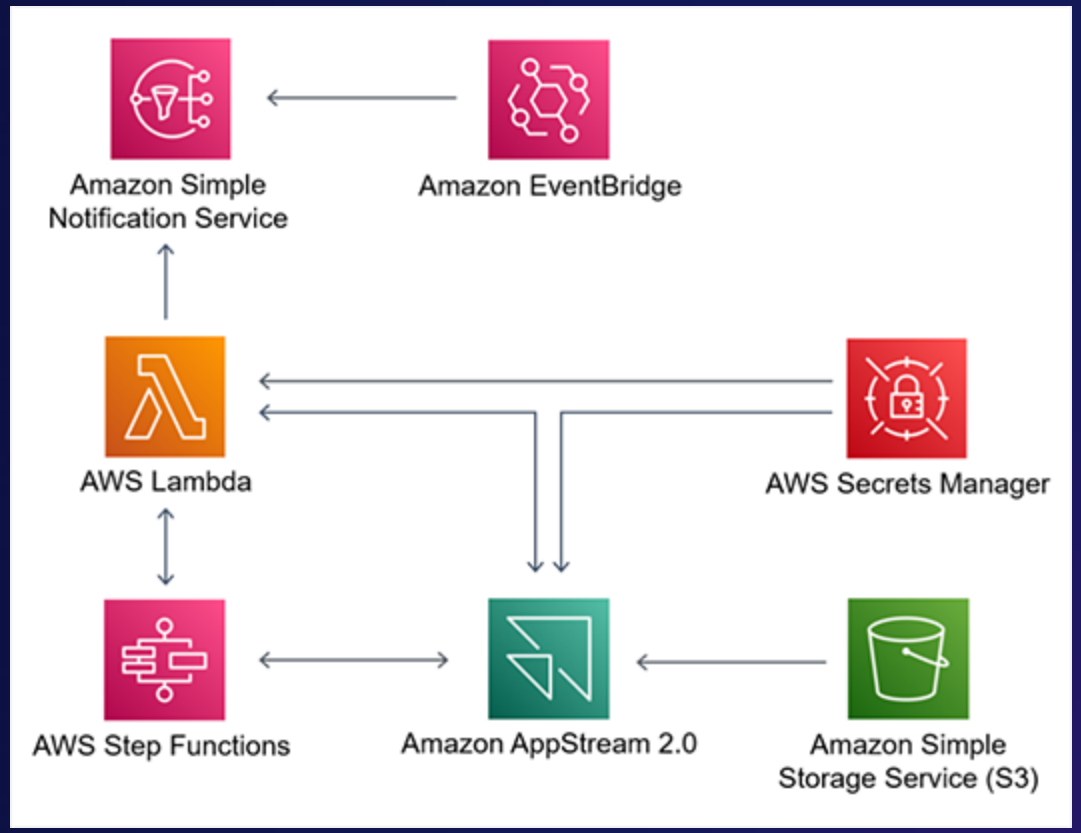
AppStream 2.0 with AWS Systems Manager

APPSTREAM



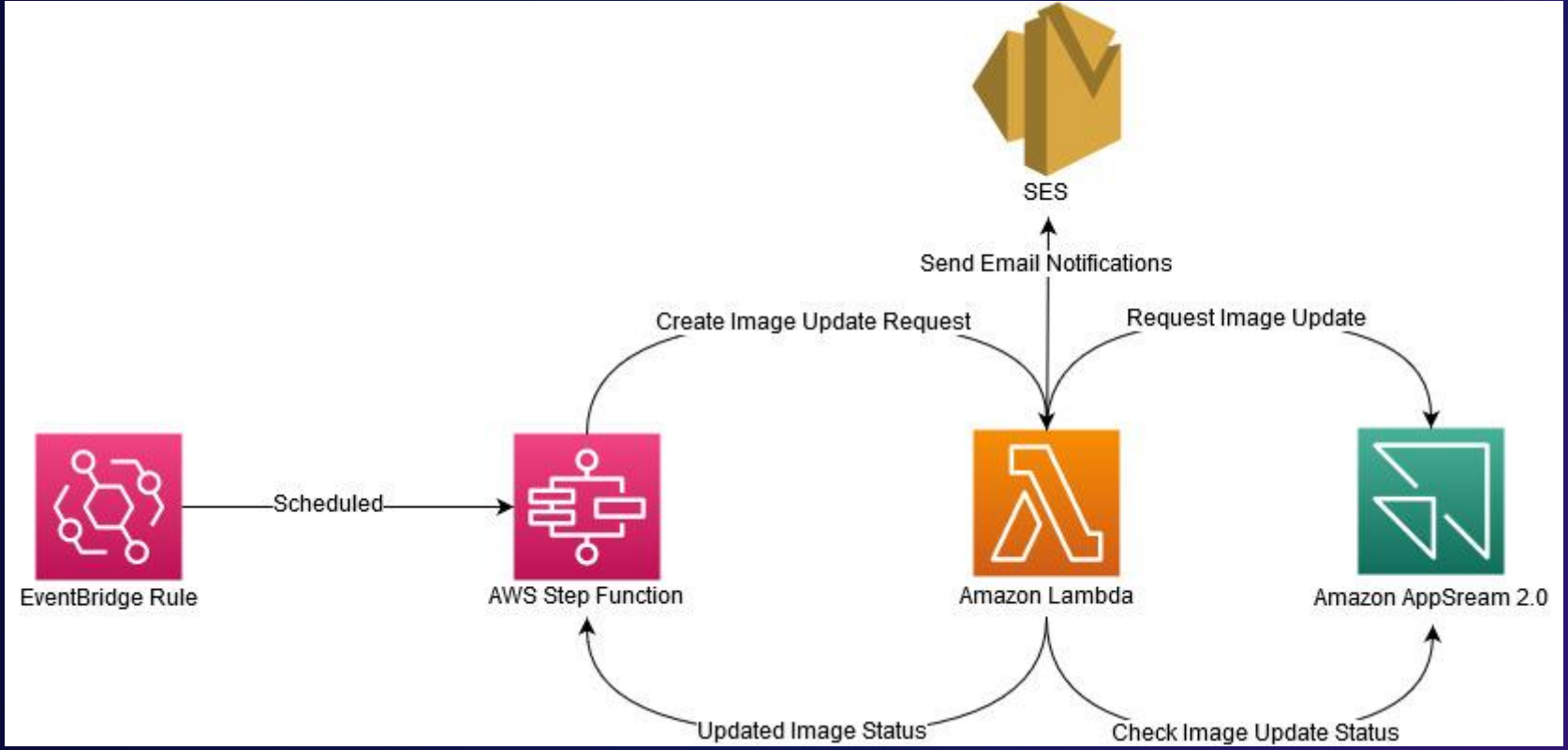
CI/CD pipelines Amazon AppStream 2.0

RELEASE

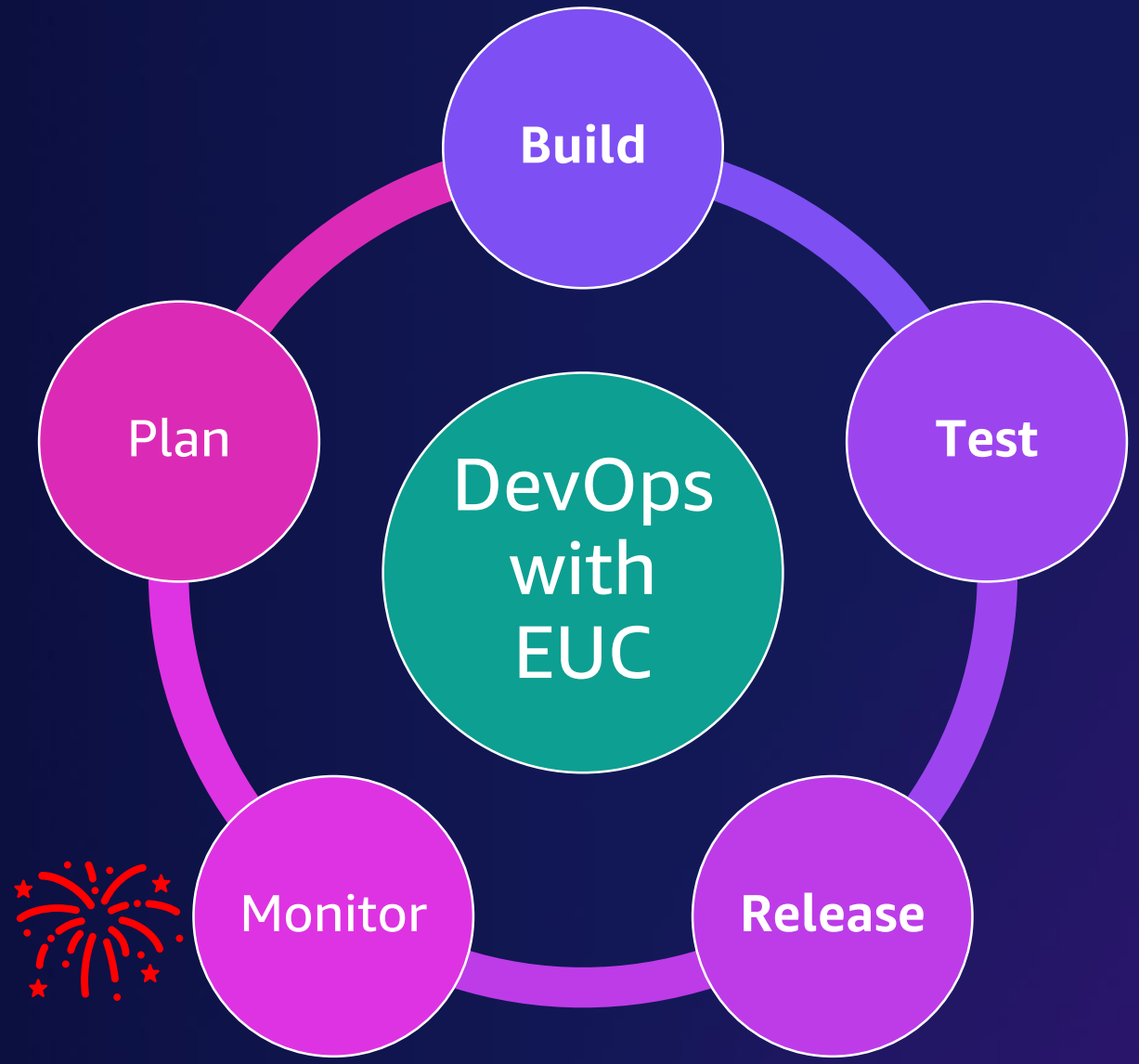


Scheduling image updates for AppStream 2.0

RELEASE



Monitor, and feedback



CloudWatch WorkSpaces Metrics

MONITOR

Metric	Description
ActualCapacity	Total number of instances that are available for streaming or are currently streaming. ActualCapacity = AvailableCapacity + InUseCapacity Units: Count
AvailableCapacity	Number of idle instances currently available for user sessions. AvailableCapacity = ActualCapacity – InUseCapacity Units: Count
CapacityUtilization	Percentage of instances in a fleet that are being used, using the following formula. CapacityUtilization = (InUseCapacity/ActualCapacity) * 100Monitoring this metric helps with decisions about increasing or decreasing the value of a fleet's desired capacity.
DesiredCapacity	Total number of instances that are either running or pending. This represents the total number of concurrent streaming sessions your fleet can support in a steady state. DesiredCapacity = ActualCapacity + PendingCapacity. Units: Count
InUseCapacity	Number of instances currently being used for streaming sessions. One InUseCapacity count represents one streaming session. Units: Count
PendingCapacity	Number of instances being provisioned by AppStream 2.0. Represents the additional number of streaming sessions the fleet can support after provisioning is complete. When provisioning starts, it usually takes 10-20 minutes for an instance to become available for streaming.
RunningCapacity	Total number of instances currently running. Represents the number of concurrent streaming sessions that can be supported by the fleet in its current state. This metric is provided for Always-On fleets only, and has the same value as the ActualCapacity metric. Units: Count
InsufficientCapacity Error	Number of session requests rejected due to lack of capacity. You can set alarms to use this metric to be notified of users waiting for streaming sessions. Units: Count

AppStream CloudWatch Monitoring

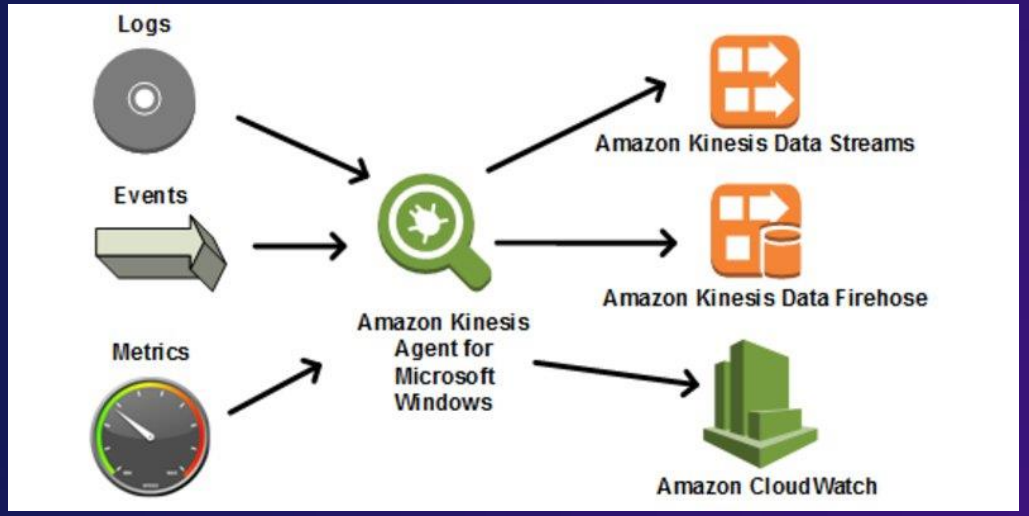
MONITOR

Metric	Description
ActualCapacity	Total number of instances that are available for streaming or are currently streaming. ActualCapacity = AvailableCapacity + InUseCapacity Units: Count
AvailableCapacity	Number of idle instances currently available for user sessions. AvailableCapacity = ActualCapacity – InUseCapacity Units: Count
CapacityUtilization	Percentage of instances in a fleet that are being used, using the following formula. CapacityUtilization = (InUseCapacity/ActualCapacity) * 100Monitoring this metric helps with decisions about increasing or decreasing the value of a fleet's desired capacity.
DesiredCapacity	Total number of instances that are either running or pending. This represents the total number of concurrent streaming sessions your fleet can support in a steady state. DesiredCapacity = ActualCapacity + PendingCapacity. Units: Count
InUseCapacity	Number of instances currently being used for streaming sessions. One InUseCapacity count represents one streaming session. Units: Count
PendingCapacity	Number of instances being provisioned by AppStream 2.0. Represents the additional number of streaming sessions the fleet can support after provisioning is complete. When provisioning starts, it usually takes 10-20 minutes for an instance to become available for streaming.
RunningCapacity	Total number of instances currently running. Represents the number of concurrent streaming sessions that can be supported by the fleet in its current state. This metric is provided for Always-On fleets only, and has the same value as the ActualCapacity metric. Units: Count
InsufficientCapacityError	Number of session requests rejected due to lack of capacity. You can set alarms to use this metric to be notified of users waiting for streaming sessions. Units: Count

Key instance metrics to monitor

MONITOR

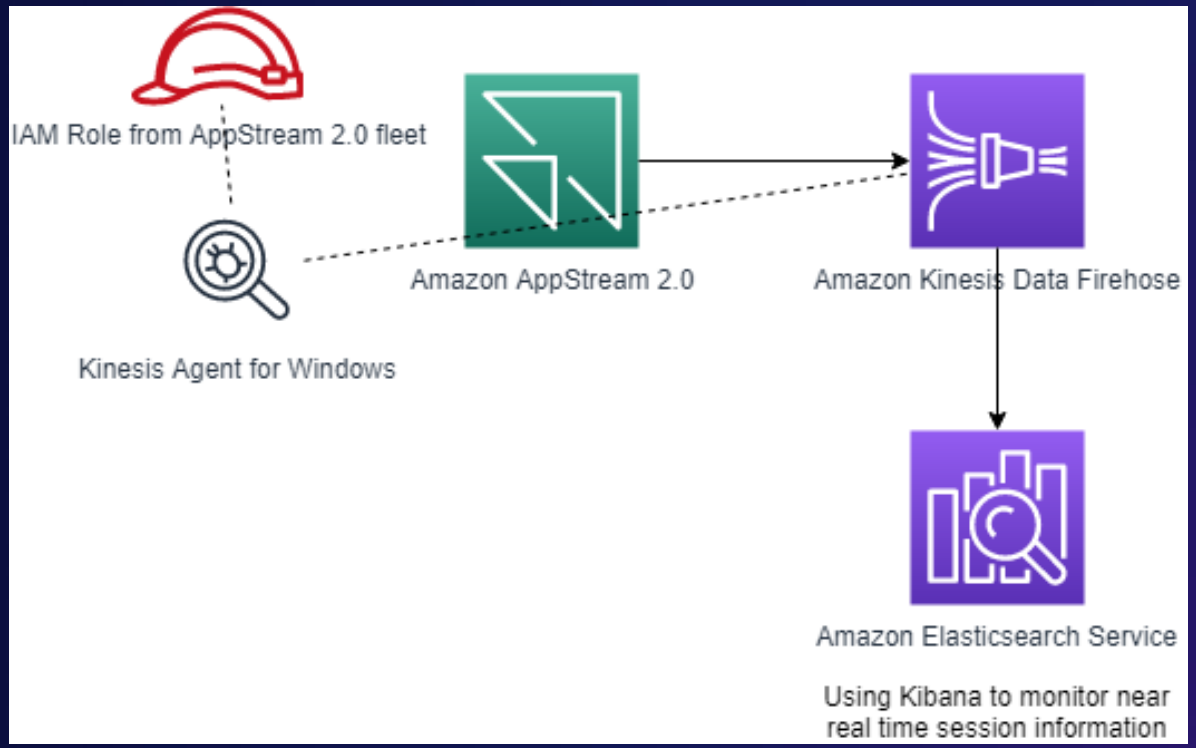
- CPU
- Memory
- Disk Storage
- Logon Duration
- Application Load Time
- Application Response Time
- Session Response Time
- Graphics Quality and Responsiveness
- Logon/Logout times



Logging Non-Persistent Sessions

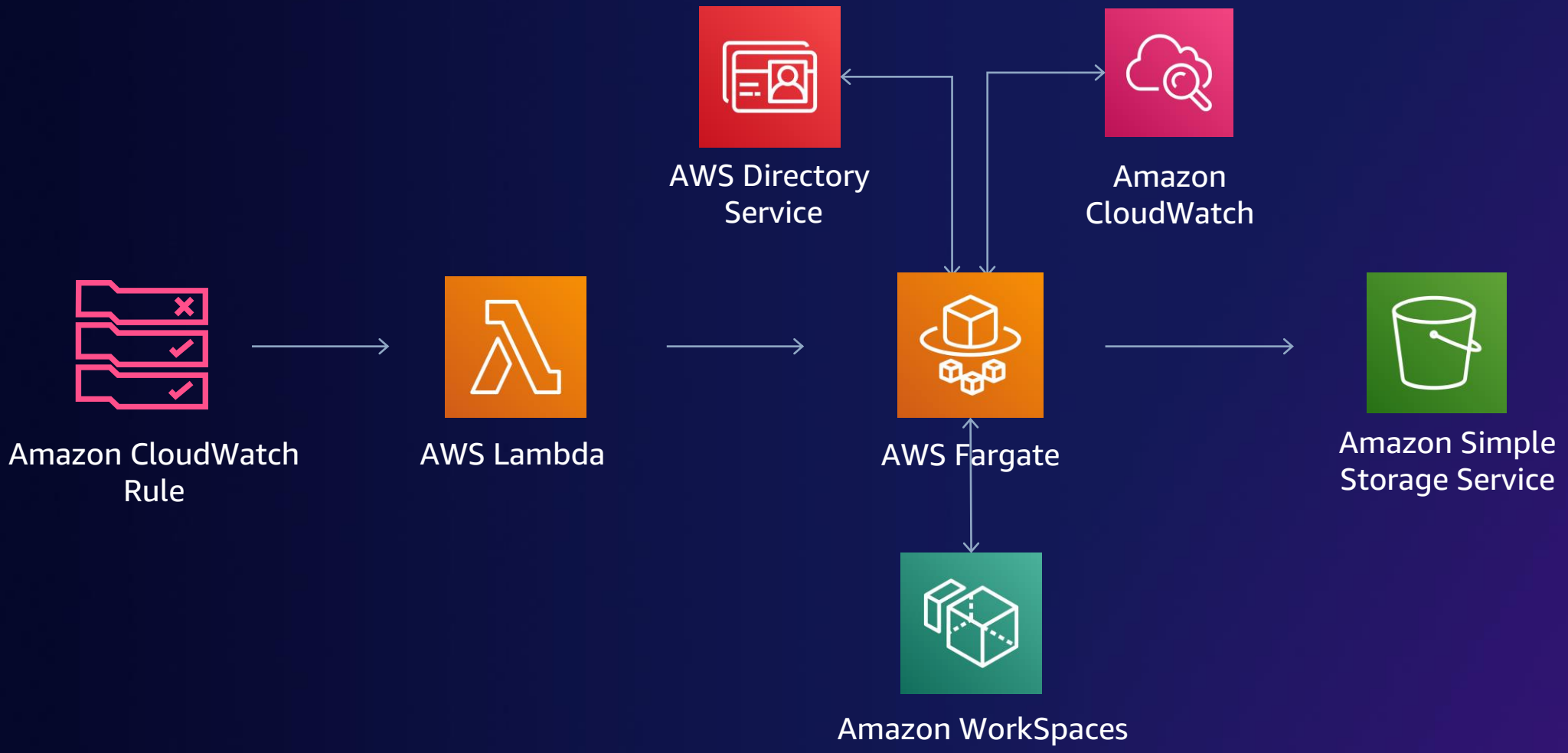
MONITOR

- Log AppStream Windows Event Logs to S3
- Log with Amazon Elastic Search and Kinesis Data Firehose



Amazon WorkSpaces Cost Optimiser

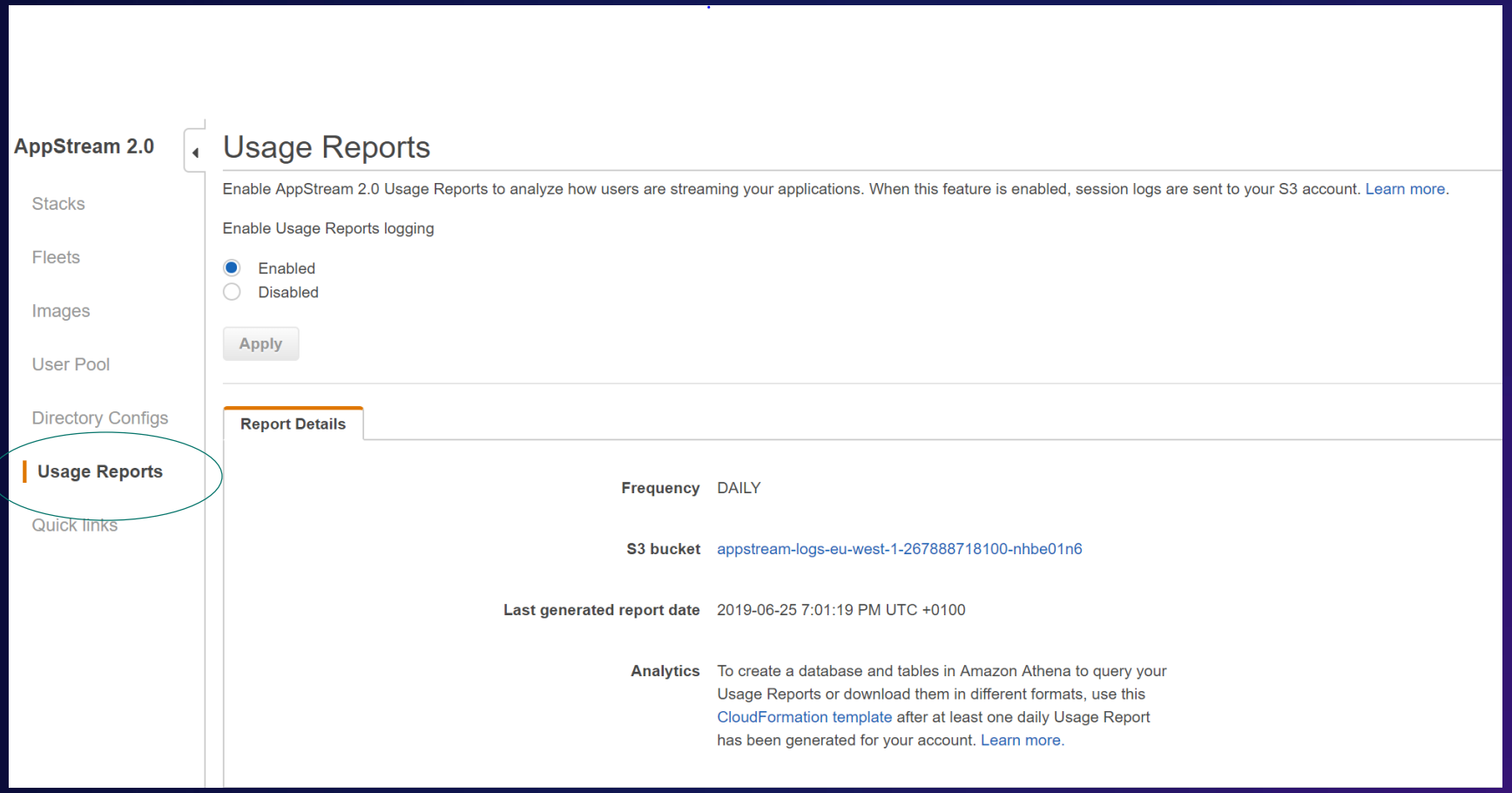
MONITOR



https://aws.amazon.com/solutions/amazon-workspaces-cost-optimizer/

AppStream Usage Monitoring

MONITOR



https://docs.aws.amazon.com/appstream2/latest/developerguide/enable-usage-reports.html

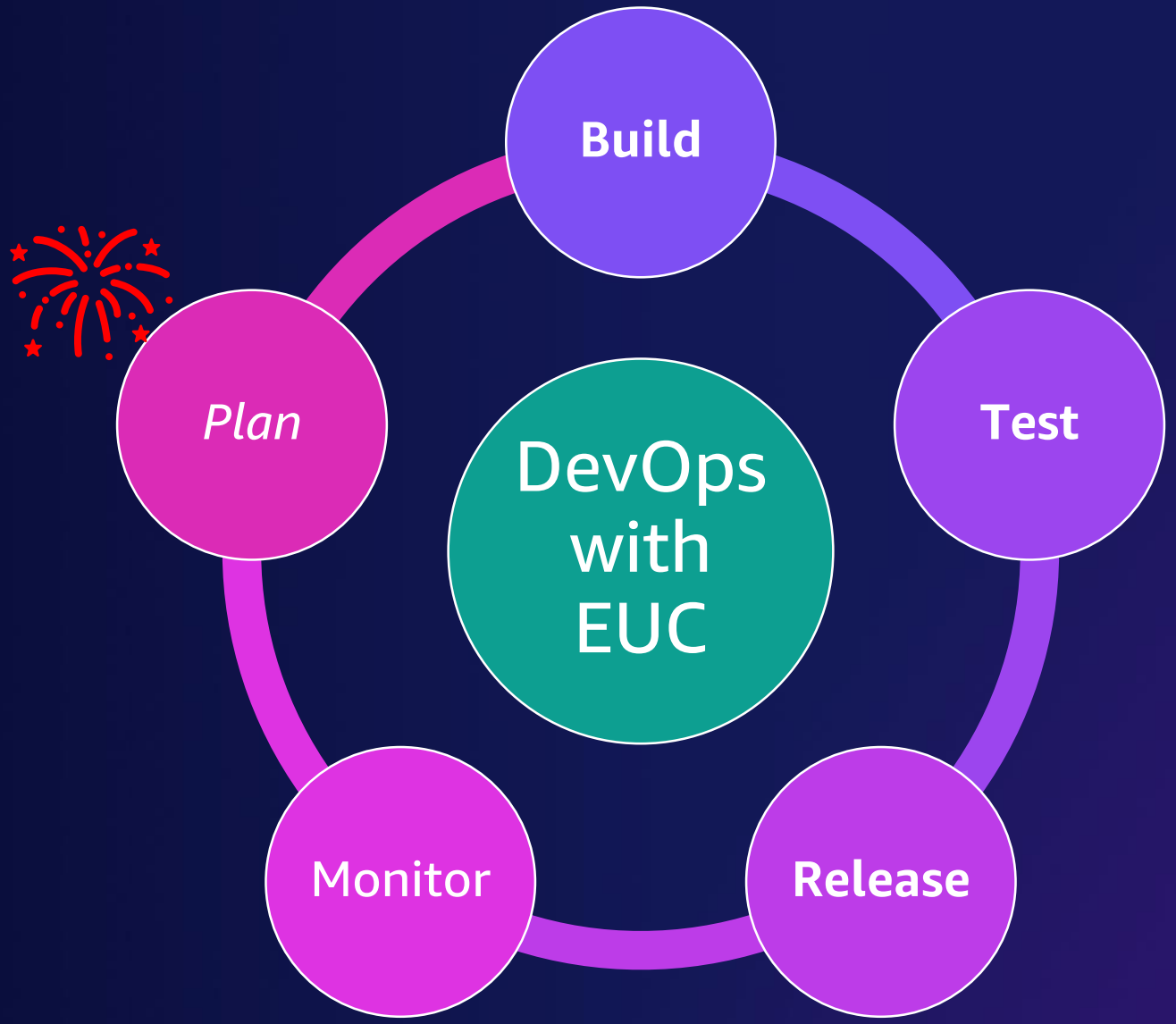
Customer feedback

MONITOR

"*The service would run perfectly if it wasn't for the users*" Anne Admyn

Combine the technical metrics with subjective information from service users

Plan



Planning for scale

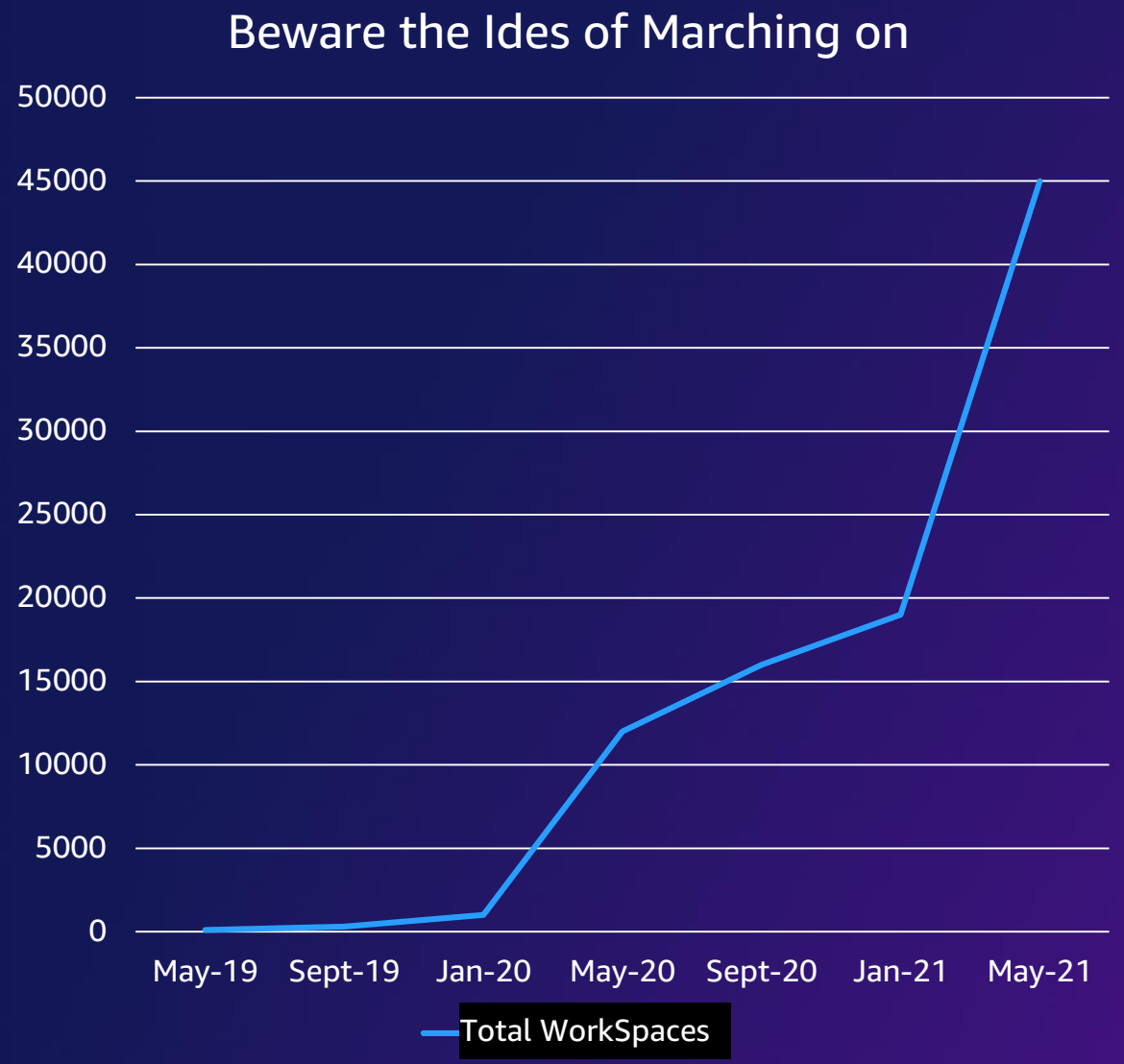
PLAN

Consider scale from the start;

You won’t always have the luxury of scaling slowly

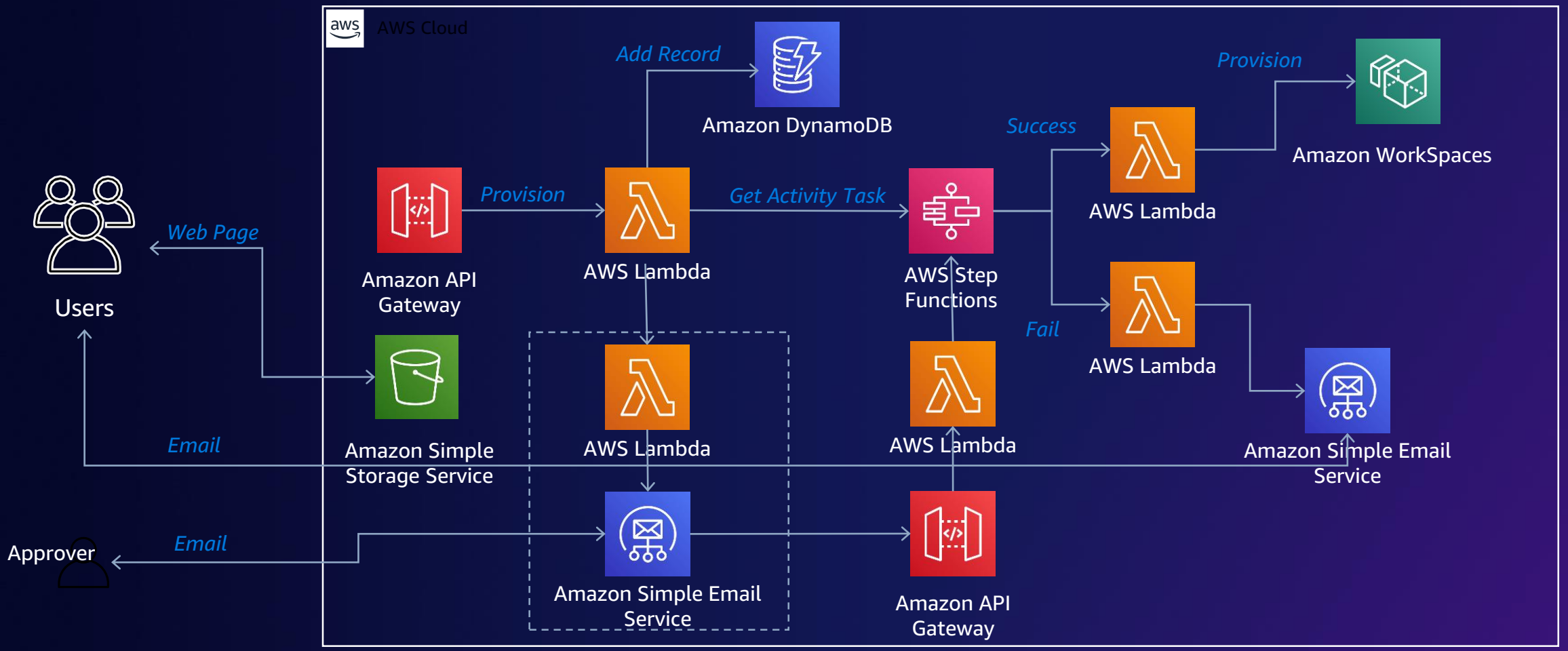
Consider deployment patterns for different user profiles

- IT administrators
- Standard users
- Development users
- Secure browser users



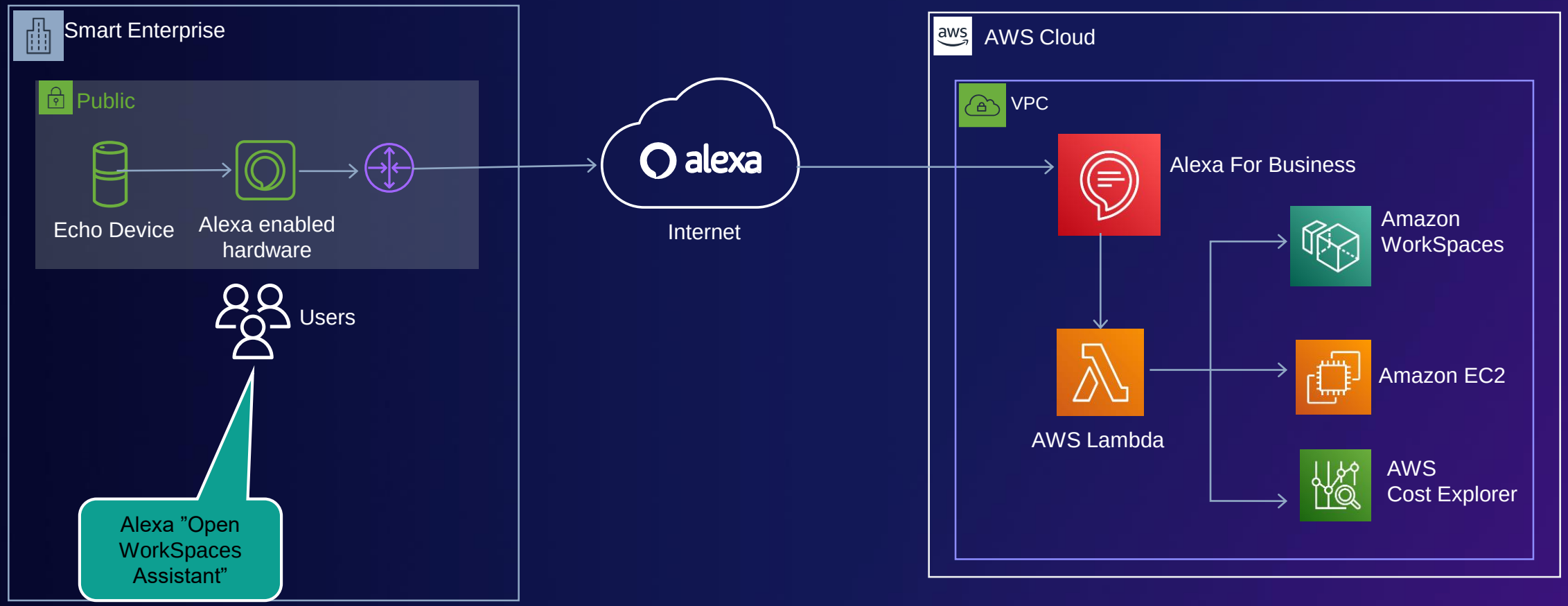
Amazon WorkSpaces with a Self-Service Portal

PLAN

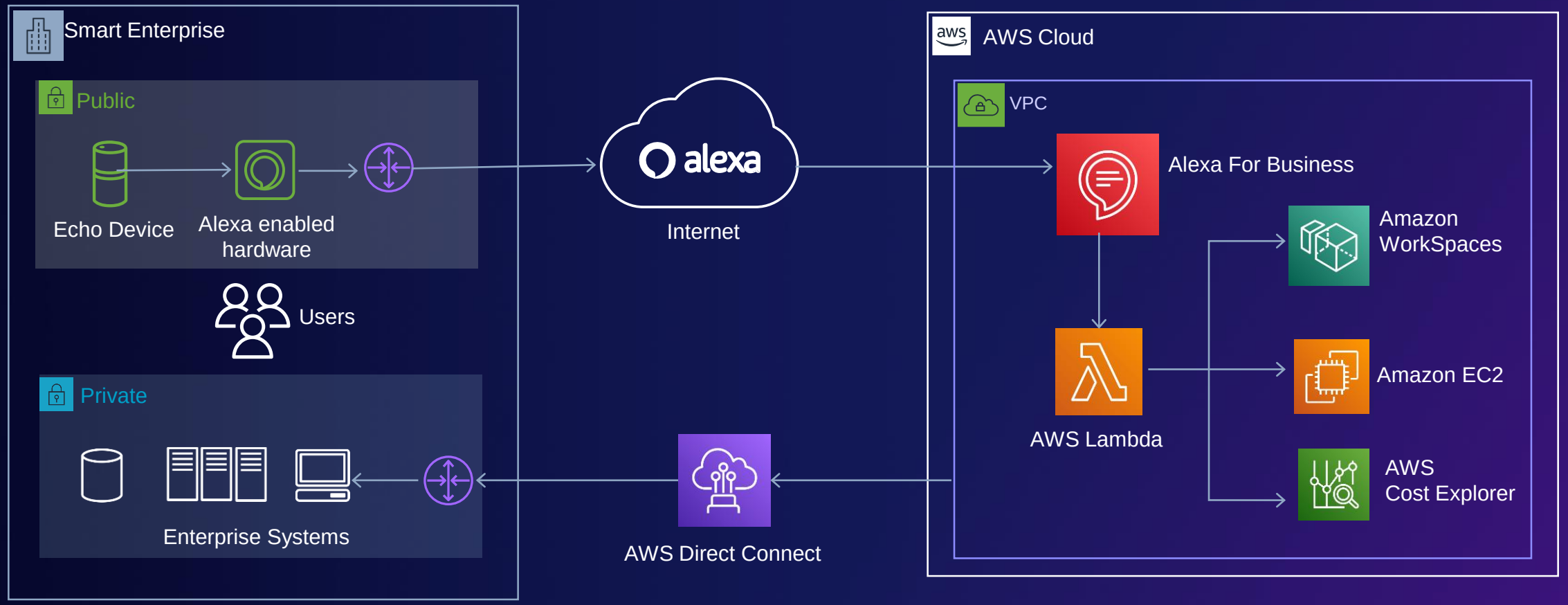


https://aws.amazon.com/blogs/desktop-and-application-streaming/automate-amazon-workspaces-with-a-self-service-portal/

Architecture for WorkSpaces Assistant



Architecture for WorkSpaces Assistant



With DevOps and Amazon End User Compute

More **distributed** and **flexible** workforce

respond to demand for services at scale

Need to improve **productivity** while reducing **cost**

Monitor to right size delivery for usage profile

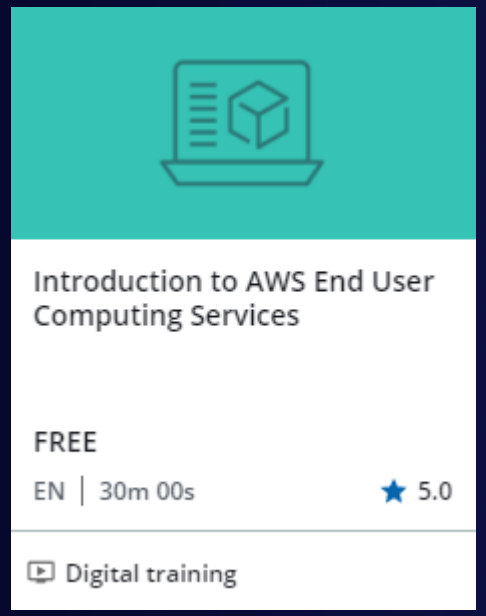
While raising the bar on **security** and **reliability**

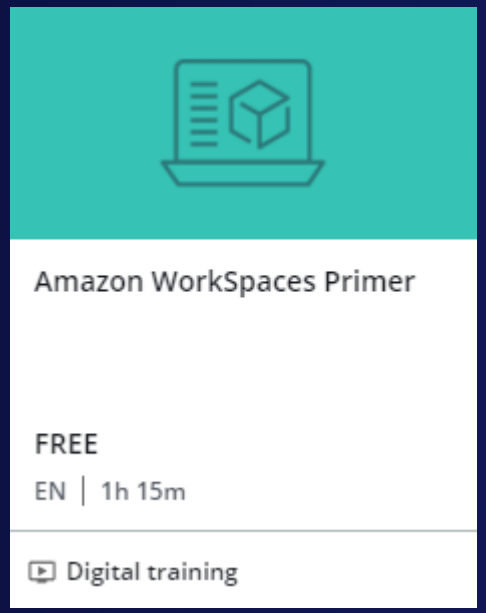
Automate, maximise confidence

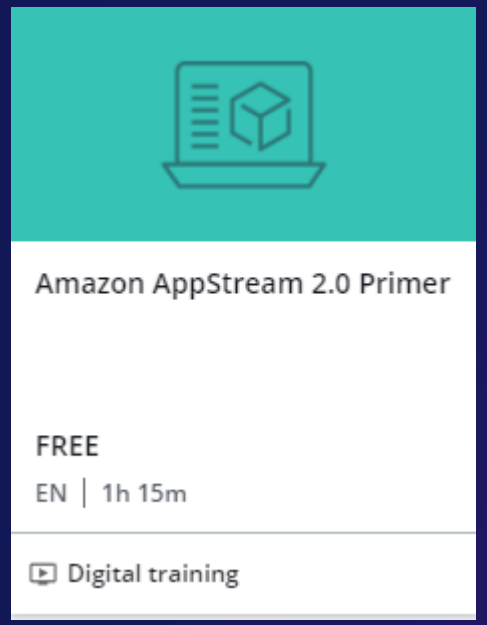
Next steps

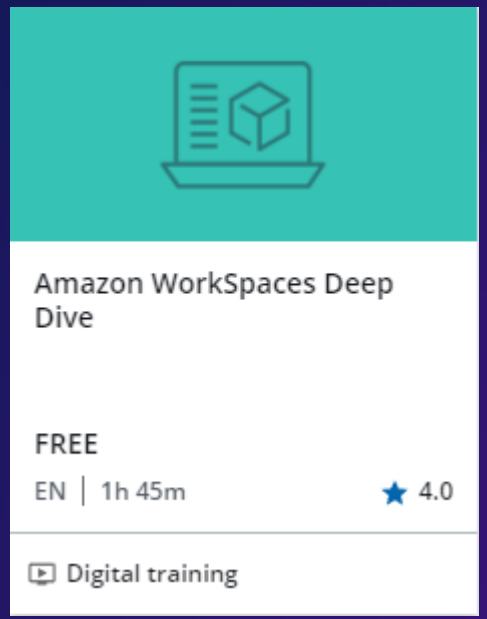
1. Schedule an AWS EUC deep dive on your use case
2. Connect with an AWS Digital Workplace Competency Partner
3. Schedule an AWS EUC Immersion Day with your team
4. Check out our free Amazon WorkSpaces and AppStream 2.0 training courses on AWS Skill Builder:

EUC courses on AWS Skill Builder









https://skillbuilder.aws

Domain: End User Computing

Learn in-demand AWS Cloud skills

AWS Skill Builder

Access **500+ free** digital courses and Learning Plans

Explore resources with a variety of skill levels and **16+** languages to meet your learning needs

Deepen your skills with digital learning on demand

Train now

AWS Certifications

Earn an industry-recognized credential

Receive Foundational, Associate, Professional, and Specialty certifications

Join the **AWS Certified community** and get exclusive benefits

Access **new** exam guides

Thank you!

Andrew Wood

@andymwood

Please complete the session survey