1. Abstract 1

2. Overview 2

3. AWS Security Concepts 3

4. Security is a Shared Responsibility 4


6. Use Cases 6

   6.1 Additional Security for Accessing AWS 6

      6.1.1 Secure AWS Root Account 6

      6.1.2 Federate Identities 7

      6.1.3 Control AWS Role Assignment, and Grant the Right Amount of Privilege 8

   6.2 Secure Privileged Access on Amazon EC2 Instances 8

      6.2.1 Extend Enterprise Authentication to Amazon EC2 Instances 8

      6.2.2 Secure Shared Accounts and Remote Access, Constrain Rights, Audit Everywhere 9

      6.2.3 Enforce Multi-factor Authentication 10

   6.3 Enable Enterprise Identities to Access Apps Hosted on AWS 10

      6.3.1 Extending Enterprise Identities to Hosted Apps 10

      6.3.2 Enforcing Multi-factor Authentication 10

7. Conclusion 11

8. Appendix 12
1. Abstract

This white paper is intended for existing and potential Centrify customers, who are seeking to expand their current Identity and Access Management infrastructure to workloads running in Amazon Web Services (AWS). It provides security best practices that can help you define controls, policies and processes to protect your data and assets in the AWS Cloud.

In this paper, we focus on best practices that are relevant to Privileged Access Management (PAM) and describe how to implement them with Centrify Infrastructure Services. We also expand on security nuances of the hybrid cloud model that do not directly map to a pure on-premises (or a pure cloud) model.

The paper is targeted at IT decision makers and security personnel and assumes that you are familiar with basic security concepts in the areas of networking, operating systems, data encryption and operational controls.
2. Overview

Migrating your applications and IT assets to AWS can help you reduce costs, become more agile and deliver products as well as services to your customers faster. According to AWS, migrating at scale to AWS calls for a level of business transformation in order to fully realize the numerous benefits of operating in a cloud environment. This might include changes to tools, processes and skillsets. As such, establishing a security foundation in the cloud that is comparable to your on-premises environments is key to the successful transition to AWS.

The main goal of this white paper is to show how Centrify can help you leverage your existing Identity and Access Management infrastructure and extend your PAM policies into AWS. The best practices outlined here will help secure access to your AWS resources, gain visibility into privileged user activity on your Amazon EC2 instances, and prevent data breaches involving weak or stolen credentials.

After reading this paper, you will become familiar with the basic AWS security concepts, the AWS Shared Responsibility Model, core security best practices that are broadly applicable to both your on-premises and cloud environments and ways to increase security in your AWS environments.
3. AWS Security Concepts

To implement Identity and Access Management controls, you must be familiar with a few key AWS security concepts. This section provides a primer on the security concepts we will refer to throughout this white paper. To learn more about AWS security fundamentals, please visit https://aws.amazon.com/security/.

AWS Security Credentials

When you interact with AWS, you specify your AWS security credentials to verify who you are and whether you have permission to access the resources that you are requesting. AWS uses the security credentials to authenticate and authorize your requests. Different types of security credentials are used depending on how you interact with AWS.

AWS Account Root User Credentials
All AWS accounts have root user credentials (that is, the username and password of the account owner). These credentials allow full access to all resources in the account. Please note that you can’t restrict permissions for root user credentials.

AWS Identity and Access Management (IAM) Credentials
IAM for AWS enables you to create and manage users, security credentials such as passwords, access keys, and permissions policies that control which AWS services and resources users can access.

IAM for AWS lets you create individual users within your AWS account and give them each their own username, password, and access keys. Individual users can then log into the console using a URL that’s specific to your account. You can also create access keys (which consist of an access key ID and secret access key) for individual users so that they can make programmatic calls to access AWS resources using the command line interface (CLI), the SDKs for AWS or API calls.

Amazon EC2 Credentials
You log in to an Amazon EC2 instance with a key pair (for Linux instances) or using a username and password (for Microsoft Windows instances) to perform activities at the operating system level. Please note, Amazon EC2 users exist outside IAM for AWS.

Temporary Security Credentials
You can use the AWS Security Token Service (AWS STS) to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM for AWS users can use but are short-term and not stored with the user.

Tracking AWS Account Activity and API Usage

Every action performed within your AWS infrastructure can be traced to an API call. AWS CloudTrail is a service that records API activity made on your account and delivers log files to your Amazon Simple Storage Service (Amazon S3) bucket. CloudTrail provides event history of your AWS account activity including actions taken through the AWS Management Console, SDKs for AWS, command line tools, and other AWS services.
4. Security is a Shared Responsibility

The AWS Shared Responsibility Model stipulates that you, as the customer, and AWS must work together towards common security objectives. AWS is responsible for protecting the infrastructure (e.g., the hardware, software, networking, facilities) that runs all of the services offered in the AWS Cloud, while you are responsible for securing operating systems, platforms and data. To ensure a secure global infrastructure, AWS configures infrastructure components and provides services and features you can use to enhance security. For example, the service that helps you control privilege and access is IAM for AWS. With IAM for AWS, you can control access to tasks that are performed using the AWS Management Console, the AWS Command Line Tools or service API operations using the SDKs for AWS.

Source: Amazon Web Services Shared Security Model, July 2018

Please note that AWS offers a Shared Responsibility Model for each of the different type of services they offer, including infrastructure services, container services and abstracted services. Specific services further delineate how responsibilities are shared between you and AWS.
5. Six Best Practices for Securing AWS Environments

Security controls for the cloud are not that different from those on-premises. You can leverage the Identity and Access Management policies and process that exist in your on-premises environments to set up your access control mechanisms for your AWS Account.

The following six best practices are broadly applicable to both your on-premises and cloud environments.

1. **Common security model**  Conventional security and compliance concepts still apply in the cloud. Leverage what you’ve already got for a common security infrastructure spanning on-premises and cloud resources.

2. **Consolidate identities**  Avoid additional silos of identity that expand your attack surface, increase overhead, and lead to identity sprawl. Instead of local IAM for AWS accounts and Access Keys, use centralized identities (e.g., Active Directory) and enable federated login.

3. **Ensure accountability**  Shared privileged accounts (e.g., ec2-user and administrator accounts) are anonymous. Ensure accountability by having users log in with their individual accounts and elevate privilege as required. Manage entitlements centrally from Active Directory, mapping roles and groups to AWS roles.

4. **Least privilege access**  Grant users just enough privilege to complete the task at hand in the AWS Management Console, AWS services, on Amazon EC2 instances and for access to hosted apps. Implement cross-platform privilege management for the AWS Management Console as well as Windows and Linux Amazon EC2 instances.

5. **Audit everything**  Log and monitor both authorized and unauthorized user sessions to Amazon EC2 instances. Associate all activity to an individual, and report on both privileged activity and access rights. Leverage Centrify Auditing & Reporting Service for privileged session recording as well as CloudTrail and Amazon CloudWatch to record and monitor API activity on your AWS account.

6. **MFA everywhere**  Thwart in-progress attacks and get higher levels of user assurance by consistently implementing Multi-factor Authentication (MFA). You can require MFA for accessing AWS Management Console, on login and privilege elevation for Amazon EC2 instances, when checking out vaulted passwords and when accessing hosted apps.

The above best practices also apply to various degrees in each of the major hybrid cloud use cases described in the next section.
Six Best Practices for Securing AWS Environments

6. Use Cases

Centrify supports the following use cases for increasing security in your AWS environments.

- Securing access to AWS Management Console;
- Securing access to Amazon EC2 instances; and
- Extending existing enterprise identities for accessing applications hosted on AWS (applications running on the Amazon EC2 instances).

6.1 Securing Access to AWS Management Console

Users who sign in to your AWS Management Console can access your AWS resources to the extent that you permit them. Centrify enables you to secure access to the Management Console by:

- Vaulting the AWS Account Root User Credentials and requiring Multi-factor Authentication for break-glass emergency access;
- Using federated Single Sign-on (SSO) to your AWS services to remove the need for long-lived access keys; and
- Controlling AWS role assignment and granting the right amount of privilege with centrally managed user identities.

6.1.1 Vaulting AWS Account Root User Credentials

When you sign up for an AWS account, you specify an email address and password that gets tied to your AWS root user account. The AWS root user account provides complete access to all AWS services and resources in the account.

AWS strongly recommends not using the root account for day-to-day interactions due to its absolute power. Instead, it should be used to create your first IAM for AWS user.

This account should be vaulted in the Centrify Privileged Access Service, to be checked out only in emergency situations. When such a situation arises, Centrify will enable you to grant explicitly-approved users temporary access to the AWS root user account.

To further protect your AWS root user account, you can require Centrify to enforce Multi-factor Authentication (MFA) during the credential check-out process. Optionally, or in addition, you can also configure AWS MFA to prompt for an authentication code after entering the checked out password on the AWS login page. This is a valuable defense against brute force attacks directly at the AWS Management Console and aligns with Gartner-advocated best practices.

One final note about the AWS root account. Since it’s so powerful with full rights across all AWS services, AWS recommends (and Centrify supports) creating separate AWS root accounts for each environment you need to manage. This allows you to separate administration of each environment (such as the typical dev, test, and prod.) Once created, you can use Consolidated Billing for Organizations to further minimize usage of these accounts.

Gartner advocates: "Minimally, use a multi-factor authentication tool supported by the cloud provider."

AWS provides MFA for log in to your AWS root account, which should be enforced.
Ultimately, AWS recommends\(^3\) that you never use the AWS root user account for every-day access. Instead of a single shared account, you should delegate a subset of privileges to accounts tied to individuals. Let’s discuss that in the next section.

### 6.1.2 Federating Login to AWS Services

Centrify advocates using IAM for AWS federation instead of creating IAM for AWS user accounts. Federation enables you to grant an existing user identity within your enterprise directory the appropriate access rights to any AWS service using IAM for AWS role assignments.

IAM for AWS supports SAML-based identity federation for delegated access to the AWS Management Console or AWS CLI. With identity federation, you avoid identity sprawl and having to provision and manage disparate identity silos or dealing with identity duplication or synchronization. Identity federation also removes the need for long-lived access keys, further reducing the attack surface of your AWS environment.

The roles assigned to each user in the Centrify Identity Services platform (which may be inherited from Active Directory) are mapped to AWS roles, effectively limiting what a user can do. The SAML token supplied during the authentication carries the instructions for limiting each user’s access permissions in AWS. For example, you can restrict a user’s access to the billing area in the AWS Management Console but grant unrestricted privileges for reviewing API event history in the CloudTrail.

Federation is easily enabled via SAML. It gives users a convenient dashboard of tiles for each of the target service and the added benefit of one-click Single Sign-on (SSO). You can extend federated SSO to SaaS applications hosted in AWS, as well as to custom apps.

For programmatic access to AWS services, AWS provides APIs and CLI tools. Just as with interactive login, you need to secure API access and ensure full accountability. API calls are signed using an AWS Access Key. However, long-lived keys can increase your attack surface. Instead, Centrify can obtain a short-lived access key from the AWS Security Token Service (STS). Since it has a limited life-span, your attack surface is greatly reduced and obtaining a new key upon expiration requires re-authentication with Centrify.

\(^3\)https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/grantaccess.html
6.1.3 Granting the Right Amount of Privilege with AWS Role Assignment

Access governance in AWS involves the management and audit of IAM for AWS roles that provide access rights to your AWS resources. When you create IAM for AWS roles, follow the standard security advice of granting least privilege — that is, granting only the permissions required to perform a task. Determine what users need to do and then create roles for them that let the users perform only those tasks.

Once IAM for AWS roles are created, you can map them to your existing Centrify roles. This enables you to leverage the Identity and Access Management policies and processes that exist in your on-premises environments for your AWS resources. For example, you can add any IT user or group from any connected directory service as a member of a Centrify role. The SAML token provided during the user authentication process will contain the Centrify role, which will be mapped to an IAM for AWS role. Thus, you will be able to leverage your existing Centrify infrastructure (Centrify Zones and role management) to centrally and granularly manage AWS user rights.

6.2 Securing Privileged Access on Amazon EC2 Instances

Accessing Amazon EC2 instances requires a key pair (for Linux instances) or using a username and password (for Microsoft Windows instances) to perform activities at the operating system level. Centrify enables you to secure access to your EC2 instances by:

- Extending your enterprise authentication to Amazon EC2 instances;
- Minimizing the attack surface by protecting shared accounts and remote access, granting just enough privilege for users to perform assigned tasks and auditing privileged user activity;
- Enforcing MFA at session initiation, server login and privilege elevation to stop in-progress attacks.

6.2.1 Extending Enterprise Authentication to Amazon EC2 Instances

Access to Amazon EC2 instances needs to be properly managed and protected. Organizations need to consider both user authentication to the instance, and authorization to perform specific actions at the operating system level.

Centrify recommends extending your existing enterprise authentication to your Amazon EC2 instances. This allows you to avoid replicating or moving user credentials to the cloud or managing multiple identity silos. You can use Centrify to broker your Active Directory identities to AWS. With Centrify Identity Broker Service, Amazon EC2 users log into instances with their existing enterprise credentials. You can audit and tie all activities on Amazon EC2 instances to unique users for full accountability.

Some customers prefer to set up multiple identity sources (e.g., Active Directory, Centrify Directory, generic LDAP) to separate customer, contractor, or outsourced IT identities from employee identities. Centrify Identity Broker Service enables you to use credentials from any of these sources to access your Amazon EC2 instances. With Active Directory, however, you get the added benefit of leveraging your Centrify Zone model to govern access.

The Centrify Identity Broker Service is designed to strengthen the security of your hosted infrastructure. It eliminates the need for direct external connectivity as all outbound traffic is tunneled through HTTPS to the Centrify platform. There’s no need to open inbound firewall ports or establish a site-to-site VPN for access to Active Directory. For example, you can use separate identity sources to authenticate users to your Amazon EC2 instances without setting up direct connectivity between your Amazon EC2 instances and these identity sources.
Additionally, Centrify’s architecture accommodates more complex scenarios across multiple Amazon Virtual Private Clouds (VPCs) in different AWS Regions and Availability Zones. With Centrify, you don’t have to establish a VPN connecting back to the enterprise for each of your Amazon VPCs.

As with Secure Access to the AWS Management Console, governance and role management is managed centrally through Centrify Infrastructure Services. Through the least-privilege model, admins log into Amazon EC2 instances as themselves and based on their Centrify roles, can elevate privileges to perform actions consistent with their job function.

6.2.2  Protecting Shared Accounts and Remote Access, Constrain Rights, and Audit Privileged Activity

For authentication, AWS bootstraps a default admin account for your new instance (e.g., ec2-user on Linux and administrator on Windows.) The Amazon EC2 Key Pair specified during the instance creation is used to access the instance, but the method is different for each platform.

For Linux, AWS installs the public key of the Amazon EC2 Key Pair as an “authorized key” for the ec2-user account. Thus, you can present the private key of the pair in an SSH connection to the Linux instance using your SSH client. For example, `ssh –i myprivatekeyfile ec2-user@publicipaddress`.

For Windows, AWS encrypts the administrator password with the public key. Through the Amazon EC2 Console, you request the password, and you’re prompted for the corresponding private key, which AWS uses to decrypt the password and display in plaintext. Going forward, you RDP to the system with the administrator account and plaintext password.

Since these are highly privileged accounts, the best practice is not to share them or use them routinely. Centrify recommends using administrator accounts for emergency access only, vaulting the account password in the Centrify Privileged Access Service, and rotating the password frequently. Instead, you can use individual enterprise identities to log into your Amazon EC2 instances and perform routine tasks.

Admins can log in to Amazon EC2 instances directly via SSH with their individual, low-privileged accounts. Alternatively, they can log into the Privileged Access Service and, based on their role, log in remotely to an instance using their enterprise credentials. Their activities can then be audited using Centrify’s session-recording feature at the proxy or host level.

Note that you can easily automate steps required to secure your Amazon EC2 instance with Centrify. At instance creation, you can programmatically install the Centrify agent, join the new instance to your Active Directory and enroll it in the Centrify Identity Service platform. You can also vault the bootstrapped account credentials and set up password rotation according to your security policies.

The final step in securing your Amazon EC2 instances is to constrain user rights. As with your on-premises servers, Centrify Infrastructure Services enables you to manage roles and permissions for Amazon EC2 users centrally. You can create Centrify roles with granular Amazon EC2 permissions and assign them to your users using least privilege as your guiding principle. This will ensure users can perform their assigned tasks and require explicit approvals for privileged access.

Access authorization is equally important for non-human accounts. Administrators can create batch files and scripts that run automatically (e.g., cron jobs and PowerShell scripts) to perform administrative tasks. Hackers and malware can target plaintext passwords embedded in these
scripts. Credentials for these accounts can then be used to gain elevated privileges and allow lateral movement within your network. Centrify recommends replacing passwords in automated scripts with API calls to the Centrify Infrastructure Services to request the passwords in real-time. Once used, the password can be rotated, mitigating any potential exposure.

6.2.3 Enforcing Multi-Factor Authentication

Some actions and tasks may be considered more sensitive than others, requiring additional user validation for a higher level of assurance. For example, you can require users accessing production Amazon EC2 instances to validate their identity using an out-of-band factor such as a code sent to a pre-enrolled device. Requiring additional user validation is a best practice and can prevent data breaches involving weak or stolen credentials.

Centrify’s MFA can be set up as a requirement for accessing your Amazon EC2 instances. Centrify also allows you to establish behavioral profiles for your Amazon EC2 users. Centrify’s analytics engine continuously analyzes user activity for anomalies against these profiles. If an activity is deemed anomalous, a second factor can be requested. The analytics approach is adaptive and does not rely on static policies. With adaptive MFA, you can secure your Amazon EC2 instances and enhance end user experience.

6.3 Enterprise Access to Hosted Apps

Centrify enables you to secure internal and external user access to your apps hosted on AWS by:

- Extending enterprise identities to hosted applications through simplified federated authentication for employees, business partners, and customers; and
- Enforcing Multi-factor Authentication (MFA) to satisfy stringent security requirements and prove compliance.

6.3.1 Extending Enterprise Identities to Hosted Apps

As a company developing hosted applications in AWS, you can SAML-enable them for federated Single Sign-On (SSO) by using one of the many SAML toolkits (e.g., C#, Ruby on Rails, Python (Bottle); PHP; and Java SAML toolkits.) Federated SSO provides stronger security than using passwords, enables timebound access with short-lived tokens, and eliminates the need to move or replicate identities within your applications or AWS.

Once your apps have been SAML-enabled, you can establish a trust relationship between the app and Centrify Application Services as the Identity Provider (IDP). Centrify enables you to integrate new applications and represent them as “tiles” on a user portal to provide 1-click SSO for the end user. A user would simply log into the Centrify Application Services Portal with their enterprise identity (e.g., in Active Directory) and click on the appropriate tile to be logged into the hosted app. Like previous examples, a privileged access request workflow can be used to obtain approval to access a specific app.

6.3.2 Enforcing Multi-Factor Authentication

You can apply Multi-Factor Authentication (MFA) upon the login to the Centrify user portal, as well as upon login to an individual hosted app (clicking on the app tile in the portal). Organizations that require stronger authentication can also take advantage of Centrify’s ability to support PKI-based authentication, which can be used to enforce smart card or derived credentials, as well as corporate device-only authentication requirements.
7. Conclusion

As enterprises are evolving, production workloads and applications are shifting to the cloud. To minimize the additional investments in time, resources, and overhead you can leverage the processes and technologies you already have in your on-premises environments to secure your cloud-based resources. This will allow you to consolidate identities, centralize identity and privilege management, control access, and audit all privileged activities across your enterprise.

The best practices outlined in this white paper can help you transition to the AWS cloud securely, cost-effectively, and with minimum disruption.

Think Secure AWS Environments. Think Centrify.
### 8. Appendix

This appendix documents a subset of best practices from Amazon’s AWS Security Best Practices document (August 2016) that are relevant in the context of Centrify Privileged Access Security. The “Centrify’s Recommendations” column documents how each can be addressed or expanded with Centrify Infrastructure Services.

#### AWS Shared Responsibility Model

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>You as the customer are responsible for the security of Amazon Machine Images (AMIs), Operating systems, Applications, Credentials, Policies and Configurations.</td>
<td>Centrify provides the means necessary to secure your layers of the AWS stack. Centrify solutions work with AWS security capabilities for a consistent management experience.</td>
</tr>
<tr>
<td>If you have higher security requirements, you can implement alternative authentication mechanisms, including LDAP or Active Directory authentication, and disable Amazon EC2 key pair authentication.</td>
<td>Centrify Identity Broker enables this higher security by extending your Active Directory, generic LDAP, and Centrify Cloud directories to AWS transparently.</td>
</tr>
<tr>
<td>You are responsible for managing your data (including classifying your assets) and for using IAM for AWS tools to apply ACL-type permissions to individual resources at the platform level, or permissions based on user identity or user responsibility at the IAM for AWS user/group level.</td>
<td>Centrify’s RBAC model complements AWS’ IAM model. Centrify roles and rights map to AWS roles so that privileges can be centrally managed through Centrify (or through Active Directory).</td>
</tr>
<tr>
<td>IAM for AWS is configured to help ensure secure access control of AWS resources.</td>
<td>Centrify centralizes your IAM-related functions for most AWS use cases. Centrify complements IAM for AWS in situations where IAM for AWS is required (e.g., local role and policy definitions.)</td>
</tr>
</tbody>
</table>

#### Define and Categorize Assets on AWS

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-factor authentication (MFA) token is enabled to provide two-factor authentication for the root AWS account.</td>
<td>Centrify provides Multi-factor authentication (MFA) across the hybrid infrastructure. Combining Centrify MFA with AWS MFA for the AWS root account results in a comprehensive MFA coverage.</td>
</tr>
</tbody>
</table>

#### Secure Your Infrastructure

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>The best protection against resource compromise is to follow the security best practices outlined in this document. While AWS provides certain security tools to help you establish strong defenses for your cloud environment, you must follow security best practices as you would for servers within your own data center. Consistently adopt simple defense practices, such as applying the latest software patches, restricting network traffic via a firewall and/or Amazon EC2 security groups, and providing least privilege access to users.</td>
<td>Centrify allows you to extend your on-premises Centrify security model to AWS at the AWS service level and the EC2 instance level. We recommend you continue to lead with a least-privilege/privilege elevation approach. For emergency situations requiring the use of a shared account, you can vault privileged accounts and strictly control the access to these credentials through a combination of access approval workflows and MFA. Auditing and session recording should occur at the host level to ensure full visibility in the event a bastion/proxy is bypassed.</td>
</tr>
<tr>
<td>Malicious, illegal, or harmful activities that use your AWS resources violate the AWS Acceptable Use Policy and can lead to account suspension.</td>
<td>To avoid this situation, you need a comprehensive security solution purposefully build for hybrid cloud. You use Centrify on-premises for the strongest security. Extend this to AWS for blanket coverage of your expanded enterprise including end-end auditing and session recording of all privileged activity.</td>
</tr>
</tbody>
</table>

Malicious, illegal, or harmful activities that use your AWS resources violate the AWS Acceptable Use Policy and can lead to account suspension.
### Manage AWS Accounts, IAM Users, Groups, and Roles

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Best practice is to create individual IAM for AWS users for each individual that needs to access services and resources in your AWS account.</strong></td>
<td>Depending on situation, Centrify advocates stronger security using SAML-based federation and Active Directory (Kerberos) or LDAP credentials, obviating the need to duplicate identities in IAM for AWS and maintaining a centralized identity management model.</td>
</tr>
<tr>
<td><strong>Do not use root account credentials for day-to-day interactions with AWS.</strong></td>
<td>Vault the AWS root account in the Centrify Privileged Access Service and control its access aggressively for break-glass situations only. Have admins login as themselves and elevate privilege as required. For emergency situations, enable Centrify’s self-service access request service and additional identity assurance via Centrify MFA and AWS MFA.</td>
</tr>
<tr>
<td><strong>We strongly discourage the use of shared user identities, where multiple entities share the same credentials.</strong></td>
<td>Centrify recommends eliminating the AWS-bootstrapped ec2-user and local administrator shared accounts. With both Linux and Windows instances joined to Active Directory, admins can log in with their individual low-privileged accounts and elevate privilege as-required. The Linux root account can be vaulted in Centrify Privileged Access Service for emergency-break glass with the additional security of using a low-privileged local Linux account as a proxy to log in over SSH.</td>
</tr>
<tr>
<td><strong>As a best practice, users should rotate their access keys on a regular basis.</strong></td>
<td>Centrify’s best practice is to avoid the use of long-lived AWS access keys, using short-lived tokens such as SAML as a more secure alternative. Such keys do not require rotation since they have a limited time to live. Note that any shared account passwords stored in the Centrify Privileged Access Service can be automatically rotated on a customer-defined schedule.</td>
</tr>
<tr>
<td><strong>Distributing long-term credentials to each instance is challenging to manage and a potential security risk.</strong></td>
<td>Access Keys and passwords are long-lived. They should not be used (whether vaulted or not.) Instead, Centrify supports the use of stronger short-lived tokens such as SAML.</td>
</tr>
<tr>
<td><strong>Managing credentials for multiple accounts makes identity management difficult.</strong></td>
<td>Avoid Access Key sprawl. Instead of creating users and Access Keys in IAM for AWS, have users log in with existing corporate credentials (Active Directory, LDAP, or SAML.)</td>
</tr>
<tr>
<td><strong>Using IAM for AWS roles and temporary security credentials means you don’t always have to manage long-term credentials and IAM for AWS users for each entity that requires access to a resource.</strong></td>
<td>Leverage Centrify Identity Broker to support your existing identity stores (Active Directory, LDAP, Centrify Directory) instead of creating a new silo in IAM for AWS. IAM for AWS roles can still be leveraged to ensure entitlements (Centrify and/or Active Directory) are mapped to AWS entitlements.</td>
</tr>
<tr>
<td><strong>We recommend you activate MFA for your AWS account and your IAM users to prevent unauthorized access to your AWS environment.</strong></td>
<td>Centrify recommends vaulting the AWS root account password for break-glass emergency use only. If required, enforce access request/approval to check out the password. For additional identity assurance, prompt the user for a second factor to check the password out. Finally, on login at the AWS Management Console, use AWS MFA in case (e.g.) of a direct brute force attack on the AWS web site.</td>
</tr>
<tr>
<td><strong>Access keys are used to digitally sign API calls made to AWS services.</strong></td>
<td>Centrify does not recommend long-lived access keys (vaulted or otherwise.) When making an API call, Centrify can request a short-lived access key from AWS’ STS to sign the API call.</td>
</tr>
<tr>
<td><strong>You can use IAM for AWS roles by creating an identity broker that sits between your corporate users and your AWS resources to manage the authentication and authorization process without needing to re-create all your users as AWS.</strong></td>
<td>Centrify provides an Identity Broker that supports user login to Amazon EC2 Linux instances using Active Directory, LDAP, or Centrify Cloud credentials. This obviates the need to duplicate identities in IAM for AWS. Authorization at the Amazon EC2 instance OS level is managed on the host via Centrify Privilege Elevation Service.</td>
</tr>
</tbody>
</table>
### Manage AWS Accounts, IAM Users, Groups, and Roles

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM for AWS groups are a powerful tool for managing access to AWS resources.</td>
<td>Rather than AWS becoming another silo to be managed, Centrify enables a single common source (Active Directory) of identity and policy for centralized management and simpler attestation.</td>
</tr>
<tr>
<td>Using IAM for AWS roles and temporary security credentials means you don’t always have to manage long-term credentials and IAM for AWS users for each entity that requires access to a resource.</td>
<td>Centrify enables use of Active Directory Groups. These groups map to Centrify roles that in turn map to AWS roles. Thus, entitlement can be centrally managed across the hybrid enterprise from Active Directory.</td>
</tr>
<tr>
<td>You can use IAM for AWS roles by enabling IAM for AWS users from another AWS account to access resources within your AWS account. This process is referred to as cross-account access.</td>
<td>Centrify enables the use of identity federation to solve this problem. Simply add the user to a role in the Centrify Application Services. This has the net effect of centralizing identity management at the IDP instead of being split between the IDP and SP.</td>
</tr>
<tr>
<td>As a best practice, users should rotate their access keys on a regular basis.</td>
<td>Avoid Access Key sprawl. Instead of creating users and Access Keys in IAM for AWS, have users login with existing in-place credentials (Active Directory, LDAP, or SAML.)</td>
</tr>
<tr>
<td>Ensuring that users have appropriate levels of permissions to access the resources they need, but no more than that, is an important part of every ISMS.</td>
<td>Centrify Privilege Elevation Service enforces a least-privilege access control model at the host level, avoiding situations where the vault and its proxy-based command filtering is bypassed by someone attacking the instance directly.</td>
</tr>
<tr>
<td>There are two primary types of credentials associated with these identities: (1) those used for sign-in to the AWS Management Console and AWS portal pages, and (2) those used for programmatic access to the AWS APIs.</td>
<td>(1) Centrify recommends: vaulting the AWS root account for emergency checkout only; managing delegated admin accounts centrally in your existing enterprise directory (e.g., Active Directory); and having users log in via SAML-based federation. (2) Centrify can request a short-term access key from AWS STS for API requests instead of a long-lived access key.</td>
</tr>
</tbody>
</table>

### Managing OS-level Access to Amazon EC2 Instances

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the shared responsibility model, you own the operating system credentials, but AWS helps you bootstrap the initial access to the operating system.</td>
<td>Vault the bootstrapped accounts (e.g., ec2-user and administrator) and allow access for break-glass emergencies only, under access request and MFA constraints. Log in using enterprise (e.g. Active Directory) accounts.</td>
</tr>
<tr>
<td>Even with resource permissions, accidental deletion by a privileged user is still a threat (including a potential attack by a Trojan using the privileged user’s credentials), which illustrates the importance of the principle of least privilege.</td>
<td>To reduce the risk of such issues, users should login as themselves (for full accountability) with least-privilege and elevate privilege via roles as required. Centrify Privilege Elevation Service also helps prevent Pass the Hash attacks by avoiding the use of administrator hashes.</td>
</tr>
</tbody>
</table>

### Protect Data in Transit

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use SSH version 2 using non-privileged user accounts.</td>
<td>Centrify Infrastructure Services allow users to login to AWS instances using individual, non-privileged accounts and then use privilege elevation at the OS level. Traditional vault solutions encourage day-to-day login with full-privileged shared accounts.</td>
</tr>
</tbody>
</table>
## Secure Your Operating Systems and Applications

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password protect the .pem files on user machines.</td>
<td>Centrify’s best practice is to disable use of Amazon EC2 Key Pairs, using Active Directory or LDAP credentials instead to log into Amazon EC2 instances.</td>
</tr>
<tr>
<td>Disable root API access keys and secret key.</td>
<td>Long-lived access keys are not recommended (vaulted or otherwise.) When making an API call, Centrify can request a short-lived access key from AWS’ STS to sign the API call.</td>
</tr>
<tr>
<td>Give users the minimum privileges they need to carry out their tasks. That way, even if a user accidentally launches an infected executable, the impact on the instance and the wider cloud system is minimized.</td>
<td>Vaulting highly privileged shared accounts (such as ec2-user) does not prevent such mistakes. These accounts should be eliminated; users should login with individual (for full accountability) low-privilege accounts and elevate privilege via roles as required.</td>
</tr>
<tr>
<td>Use bastion hosts to enforce control and visibility.</td>
<td>Although Centrify can provide one, our best practice is to enforce control with an agent and audit on every Amazon EC2 instance. This can’t be bypassed whereas a central bastion host/proxy can.</td>
</tr>
<tr>
<td>Regularly run least privilege checks using IAM for AWS user Access Advisor and IAM for AWS user Last Used Access Keys.</td>
<td>User privileges at the host level are managed centrally in your enterprise Active Directory infrastructure. Centrify best practice is to eliminate Access Keys and leverage more secure forms of authentication such as Active Directory-based login or SAML.</td>
</tr>
</tbody>
</table>

## Manage Security Monitoring, Alerting, Audit Trail, and Incident Response

<table>
<thead>
<tr>
<th>Best Practices on AWS</th>
<th>Centrify’s Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage commands that can be used during sessions. For interactive sessions like SSH or appliance management, or AWS XLI, such solutions can enforce policies by limiting the range of available commands and actions.</td>
<td>Many solutions filter commands at a (bastion) host level. This is insufficient; it can be bypassed by directly compromising an instance and with malware on the instance. Centrify enforces access controls at the host level and so is not affected when the bastion is bypassed.</td>
</tr>
<tr>
<td>Restrict leapfrogging or remote desktop hopping by allowing access only to target systems.</td>
<td>Remote login to instances through the Centrify Secure Remote Access Service surgically places the user on the target server. It does not require use of a VPN that could expose the wider network. Centrify Infrastructure Services can restrict leapfrogging to target servers through RBAC at its portal and can prevent leapfrogging at the host/OS level including IPSec-based controls to isolate critical server-to-server communication.</td>
</tr>
<tr>
<td>Consider using a privilege escalation gateway to manage access control logs and authorization.</td>
<td>Centrify Infrastructure Services manage authorization at the host level and audit logs and recorded sessions. Gateway or proxy-based command filters can be bypassed and do not control access for (e.g.) malware running on the instance itself. If bypassed, without host-based controls, you cannot control access or record sessions.</td>
</tr>
<tr>
<td>We recommend configuring the following areas for logging and analysis: Actions taken by any individual with root or administrative privileges; access to all audit trails; invalid logical access attempts; use of identification and authentication mechanisms; initialization of audit logs; creation and deletion of system level objects. The shared responsibility model requires you to monitor and manage your environment at the operating system and higher layers.</td>
<td>Centrify audits all user activities across the hybrid enterprise and ties all activity back to a unique individual user for 100% accountability. Privileged login sessions can be video recorded either at the bastion host/proxy level or at the host level to avoid being bypassed by a direct login to the instance. All attempts to login to Centrify UIs, AWS portals, and Amazon EC2 instances as well as all privilege elevation attempts can be audited and recorded. Centrify auditing and session recording provides end-end visibility of “who has access to what” and “who did what”. It can record sessions at a proxy and OS-level. Centrify also integrates with leading commercial SIEM solutions such as Splunk, IBM QRadar, and Micro Focus ArcSight.</td>
</tr>
</tbody>
</table>
Centrify delivers Zero Trust Security through the power of Next-Gen Access. The Centrify Zero Trust Security model assumes that users inside a network are no more trustworthy than those outside the network. Centrify verifies every user, their devices, and limits access and privilege. Centrify also utilizes machine learning to discover risky user behavior and apply conditional access — without impacting user experience. Centrify’s Next-Gen Access is an industry-recognized solution that uniquely converges Identity-as-a-Service (IDaaS), enterprise mobility management (EMM) and privileged access management (PAM). Over 5,000 worldwide organizations, including over half the Fortune 100, trust Centrify to proactively secure their businesses. To learn more visit www.centrify.com.