AWS Systems Manager

AWS Certified Solutions Architect– Professional (SAP-C01)
Study notes - Sep'2019

What is SSM

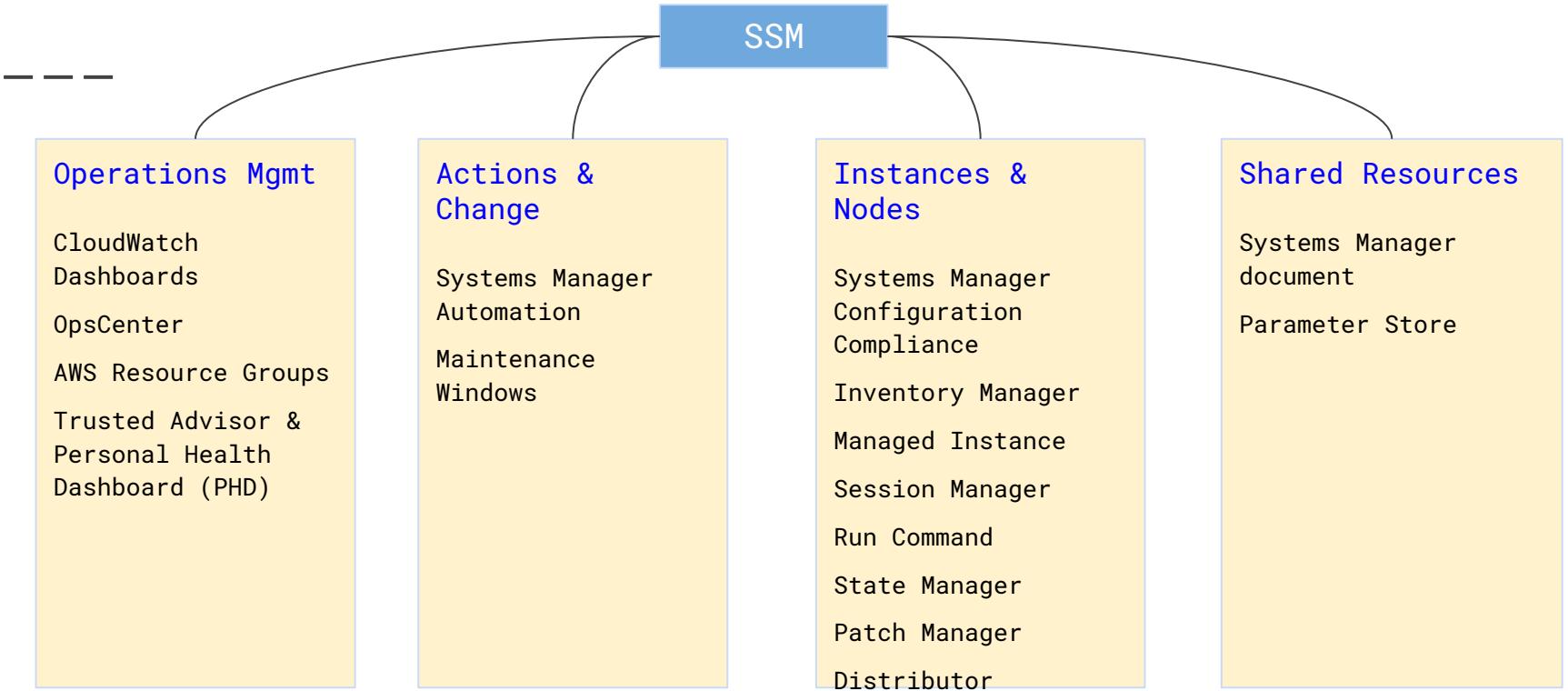
View and control your infrastructure on AWS. Using the SSM console, you can **view operational data** from multiple AWS services and **automate operational tasks** across your AWS resources. Systems Manager helps you **maintain security and compliance** by scanning your managed instances and reporting on (or taking corrective action on) any policy violations it detects

---

Example Capabilities:

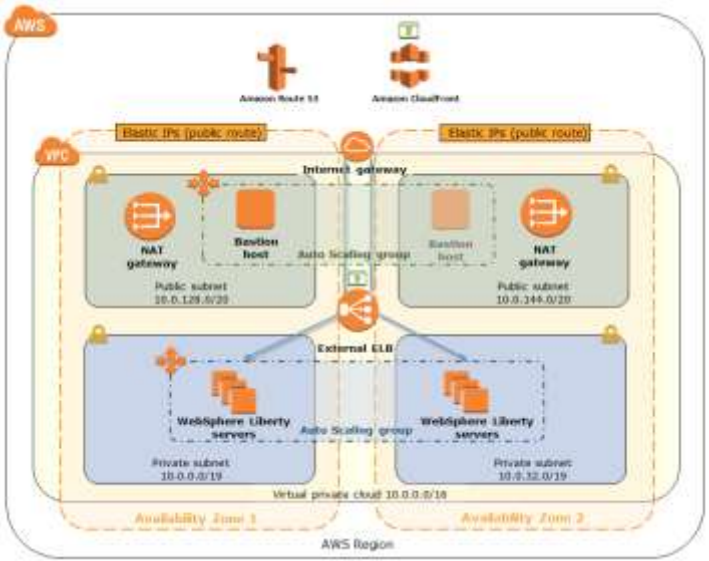
1. Group AWS resources together by any purpose or activity you choose, such as application, environment, region, project, campaign, business unit, or software lifecycle.
2. Centrally define the configuration options and policies for your managed instances.
3. Centrally view, investigate, and resolve operational work items related to AWS resources.
4. Automate or schedule a variety of maintenance and deployment tasks.
5. Use and create runbook-style SSM documents that define the actions to perform on your managed instances.
6. Run a command, with rate and error controls, that targets an entire fleet of managed instances.
7. Securely connect to a managed instance, without having to open an inbound port or manage SSH keys.
8. Separate your secrets and configuration data from your code by using parameters, with or without encryption, and then reference those parameters from a number of other AWS services.
9. Perform automated inventory by collecting metadata about your Amazon EC2 and on-premises managed instances. Metadata can include information about applications, network configurations, and more.
10. View consolidated inventory data from multiple AWS Regions and accounts that you manage.
11. Quickly see which resources in your account are out of compliance and take corrective action from a centralized dashboard.

SSM capabilities can be grouped into 4 groups



Sample EC2 stack

https://aws.amazon.com/quickstart/architecture/ibm-websphere-liberty/
https://aws-quickstart.s3.amazonaws.com/quickstart-ibm-websphere-liberty/doc/i



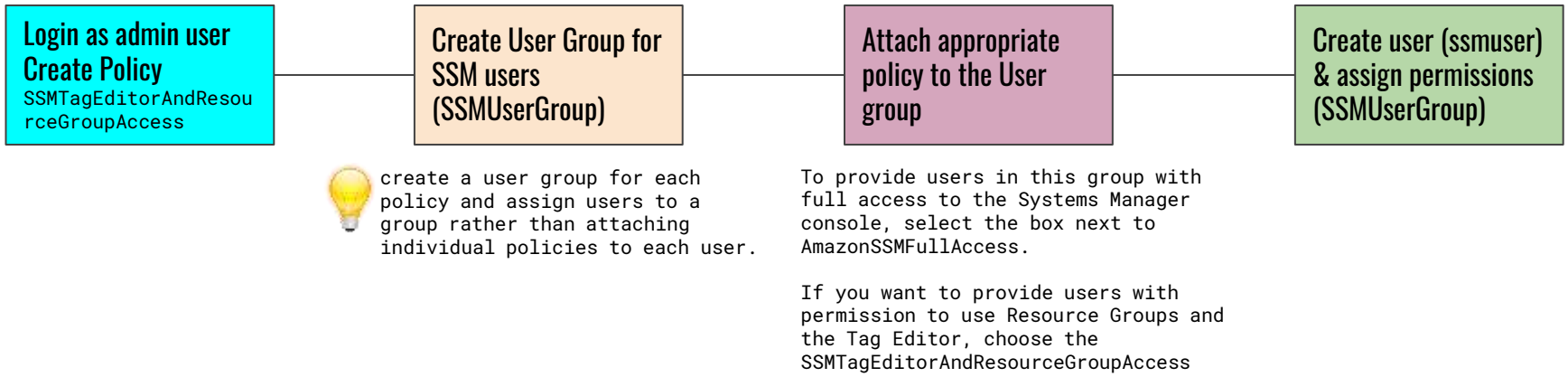
- A virtual private cloud (VPC) configured across two Availability Zones.
- In each Availability Zone, this Quick Start provisions one public subnet and one private
- subnet. This creates a logically isolated networking environment that you can connect to
- your on-premises data centers or use as a standalone environment.*
- An internet gateway to provide internet access to each subnet.*
- Managed NAT gateways deployed into the public subnets and configured with an
- Elastic IP address for outbound internet connectivity. These instances provide
- internet access for all EC2 instances launched within the private network.*
- A Linux bastion host in the public subnet to allow inbound Secure Shell (SSH) access to
- the WebSphere Liberty instances in the private subnets.*
- In the private subnets, WebSphere Liberty server instances across both Availability Zones, to ensure high availability.
- Auto Scaling enabled for the WebSphere Liberty cluster, to automatically add or remove servers based on their use, providing additional servers during peak hours and lowering costs by removing servers during off hours. This functionality is tightly integrated with the Application Load Balancer and automatically adds and removes instances from the load balancer. The default installation sets up CPU-based thresholds for scaling the instance capacity up or down. You can modify these thresholds during launch and after deployment.
- The Elastic Load Balancing service, which provides HTTP and HTTPS load balancing across the WebSphere Liberty instances. This Quick Start uses an Application Load Balancer, which is configured to use HTTP.
- An IAM role with fine-grained permissions for access to AWS services necessary for the deployment process.
- Appropriate security groups for each instance or function to restrict access to only necessary protocols and ports. For example, access to HTTP(S) server ports on Amazon EC2 web servers is limited to the Application Load Balancer.
- Amazon CloudFront as an optional content delivery network. This service enables caching of static content at the edge locations and results in lower latency in delivering content to end users. The edge locations include many Points of Presence (PoPs) across the globe to ensure low latency.
- In case of SSL/TLS implementation, AWS Certificate Manager (ACM) to provision certificates for the Application Load Balancer and CloudFront. If you're using the default CloudFront certificate, ACM is not required.

Systems Manager setup

Create Non-Admin IAM Users and Groups for Systems Manager

Users in the administrators group for an account have access to all AWS services and resources in that account. It is recommended to create users with permissions that are limited to AWS Systems Manager.

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.



Systems Manager setup

Create & Attach an IAM Instance Profile for Systems Manager

Best practise: To provide permissions for communication between instances and the Systems Manager API, we recommend creating custom policies that take into account your system needs and security requirements.

AWS Systems Manager doesn't have permission to perform actions on your instances
You need to create an **Instance profile**

An **Instance profile** is a container that passes IAM role information to an EC2 instance at launch

Create an **Instance profile** by attaching 1 or more IAM policies to a **Role**

After you create the instance profile, you attach it to the instances that you want to use with SSM

Step 1: Create an IAM Instance Profile for Systems Manager

Create Role "SSMInstanceProfile" → **Select type of trusted entity - EC2** → **Attach permissions:**
AmazonSSMManagedInstanceCore
Custom policy for S3 bucket (optional)
AmazonSSMDirectoryServiceAccess
CloudWatchAgentServerPolicy

Choose this role when you create new instances that you want to manage by using Systems Manager.

Step 2: Attach an IAM Instance Profile to an Amazon EC2 Instance

Prerequisite: SSM agent & TLS certificate must be installed in the EC2 instance (already present in Amazon AMIs)

In console, Launch Instance → Choose an AMI → Configure Instance details → Select the IAM role "**SSMInstanceProfile**"

Operations Management -
AWS Systems Manager OpsCenter

OpsCenter provides a central location where operations engineers can view, investigate, and resolve operational work items (OpsItems). OpsCenter is designed to reduce mean time to resolution

Features:

1. Automated and manual OpsItem creation - automatically create OpsItems for any AWS service that publishes events to CloudWatch Events. Also can be manual
2. Detailed and searchable OpsItems & Summary reports
3. Easy remediation using runbooks
4. Change notification
5. IAM Access control

OpsCenter can create CW rules for the foll. common operational events:

- EC2 Instance State-change - Stopped or Terminated
- EBS Snapshot Notification - Copy/Create/Share Failed
- AutoScaling EC2 Instance Launch/Term Unsuccessful
- EC2 SSM Maintenance Window Execution Failed/Timed Out
- AWS Health - RDS Maintenance Scheduled
- AWS Health - RDS Issue Notification (example: RDSConnectivity issue)
- AWS Health - EC2 Maintenance Scheduled
- AWS Health - EC2 Issue Notification (example:Instance Auto Recovery Failure)
- AWS Health - EBS Issue Notification (example:Degraded EBS volume performance)

Actions and Change

AWS Systems Manager Automation

Build Automation workflows to configure and manage instances and AWS resources.
Create custom workflows or use pre-defined workflows maintained by AWS.
Receive notifications about Automation tasks and workflows by using CW Events.
Monitor Automation progress and execution details by using the EC2 or SSM console.

Automation Use Cases:

1. **Perform common IT tasks. e.gs:**
   a. AWS-StopEC2InstanceWithApproval document
   b. AWS-StopEC2Instance document
   c. AWS-UpdateCloudFormationStackWithApproval document
2. **Safely perform disruptive tasks in bulk. e.gs:**
   a. AWS-RestartEC2InstanceWithApproval document for multiple instances
3. **Simplify complex tasks. e.gs:**
   a. AWS-UpdateLinuxAmi and AWS-UpdateWindowsAmi documents to create golden AMIs from a source AMI.
   b. AWSSupport-ExecuteEC2Rescue document to recover impaired instances.
4. **Enhance operations security**
   Use delegated administration. Create an IAM role with the permissions required to successfully stop and start EC2 instances. Customize the AWS-RestartEC2Instance document and embed the role in the document
1. **Share best practices**

An SSM automation document defines the Automation workflow (the actions that Systems Manager performs on your managed instances and AWS resources).

Automation Executions:

1. Running a Simple Automation Workflow
2. Running an Automation Workflow Step by Step
3. Running an Automation Workflow with Approvers
4. Running Automation Workflows That Use Targets and Rate Controls (Concurrency & Error threshold)
5. Running Automation based on Triggers
   a. CloudWatch EVENTS
   b. State Manager
   c. Maintenance Windows

You can have a maximum of 25 concurrent Automations running at one time.

AWS Systems Manager Automation

- Simplify AMI Patching Using Automation, Lambda, and Parameter Store
- Using Automation with Jenkins deployed on EC2 via CLI

💡 This example uses the model where an organization maintains and periodically patches their own, proprietary AMIs rather than building from Amazon EC2 AMIs.

Task 1: Create a Parameter in Systems Manager Parameter Store

Name: latestAmi. Value: a Windows AMI ID. For example: ami-188d6e0e.

Task 2: Create an IAM Role for AWS Lambda

Includes **AWSLambdaExecute** and **AmazonSSMFullAccess** managed policies. These policies give Lambda permission to update the value of the latestAmi parameter using a Lambda function and SSM

Task 3: Create an AWS Lambda Function

function automatically updates the value of the latestAmi parameter

Task 4: Create an Automation Document (*'UpdateMyLatestWindowsAmi'*) and Patch the AMI

You can add Automation as a post-build step to pre-install application releases into AMIs. You can also use the Jenkins scheduling feature to call Automation and create your own OS patching cadence.

Task 5: Configure IAM roles for Automation.

SSM requires an instance profile role and a service role ARN to process Automation workflows

Task 6: Create an IAM user account for your Jenkins server.

The Automation workflow uses the IAM user account's Access key and Secret key to authenticate the Jenkins server during execution. For permissions, assign *'AmazonSSMFullAccess'* policy

Task 7 : Configure Jenkins for Automation

- In Jenkins EC2, configure AWS CLI with the AWS Access key and secret key created above
- In Jenkins console, Choose project → Add build step → Execute shell command → run foll. Command:

```
aws --region region-id ssm start-automation-execution \
    --document-name UpdateMyLatestWindowsAmi \
    --parameters \
        "sourceAMIid='{{ssm:latestAmi}}'"
```

AWS Systems Manager Maintenance Windows

AWS Systems Manager Maintenance Windows **let you define a schedule** for when to perform potentially disruptive actions on your instances such as patching an operating system, updating drivers, or installing software or patches.

Each maintenance window has a:

- Schedule (dates to run or not run, in I18N time zone)
- a maximum duration
- a set of registered targets (the instances or other AWS resources that are acted upon) and
- a set of registered tasks.

Maintenance windows support running four types of tasks:

1. Systems Manager Run Command commands
2. Systems Manager Automation workflows
3. AWS Lambda functions
4. AWS Step Functions tasks

A **service-linked role** is a unique type of IAM role that is linked directly to Systems Manager. Service-linked roles are predefined by Systems Manager and include all the permissions that the service requires to call other AWS services on your behalf. Systems Manager uses the service-linked role named ***AWSServiceRoleForAmazonSSM***
Currently, only two Systems Manager capabilities use the service-linked role: **Inventory** & **Maintenance Windows**

Controlling Access to Maintenance Windows:

Task 1: Configure instance permissions

Provide the Maintenance Windows service with the IAM permissions needed to run maintenance window tasks on your instances by doing one of the following:

- Create a custom service role for maintenance window tasks
- Create a service-linked role for SSM

Task 2: Configure user permissions

Granting iam:PassRole permissions to the users in your account who assigns tasks to maintenance windows. Without this explicit permission, a user can't assign tasks to a maintenance window.

Should I Use a **Service-Linked Role** or a **Custom Service Role** to Run Maintenance Window Tasks?

Use Custom Service Role IF:

- You want to send SNS notifications
- You want More restrictive permissions (e.g. limited instances or restrict to certain SSM documents)
- You want More permissive permissions (e.g. Use CloudFormation or Create EBS snapshot)

For All other use cases, use Service-linked Role

AWS Systems Manager - Instances & Nodes

Configuration Compliance

Scan your fleet of managed instances for **patch compliance** and **configuration inconsistencies**.

Features

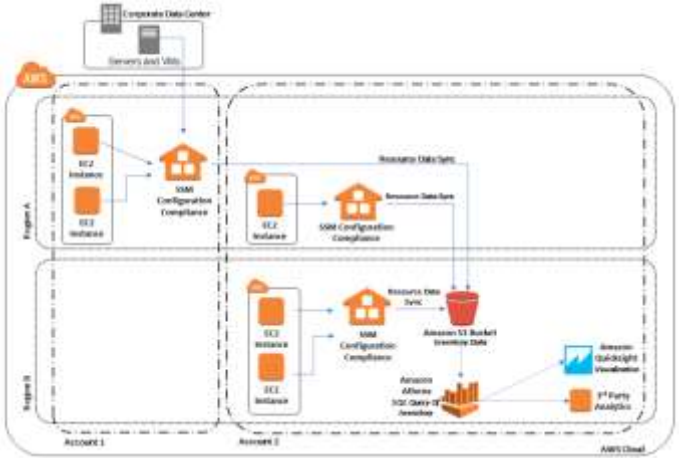
- View compliance history and change tracking for **Patch Manager** patching data and **State Manager associations** by using AWS Config.
- Customize Systems Manager Compliance to create your own compliance types based on your IT or business requirements.
- Remediate issues by using Systems Manager Run Command, State Manager, or CloudWatch Events.
- Port data to Amazon Athena and Amazon QuickSight to generate fleet-wide reports.

SSM integrates with **Chef InSpec**. InSpec is an open-source, runtime framework that enables you to create human-readable profiles on GitHub or S3. Then you can use SSM to run compliance scans and view compliant and noncompliant instances.

For viewing current compliance status, use Config Compliance
*For patching & compliance **history** & change tracking, use AWS Config*

Task: Create a Resource Data Sync for Configuration Compliance

To send compliance data from all of your managed instances to a target Amazon S3 bucket.
With all compliance data stored in a target Amazon S3 bucket, you can use services like Athena and QuickSight to query and analyze the aggregated data.



AWS Systems Manager - Instances & Nodes

Inventory

Collect *metadata* from your managed instances from multiple AWS Regions and accounts.If the pre-configured metadata types collected by Systems Manager Inventory don't meet your needs, then you can create *custom inventory*.

Metadata types:

- Applications
- AWS components
- Files
- N/w configuration
- Windows updates
- Instance details
- Services
- Tags
- Windows registry
- Windows roles
- Custom inventory

Working with File and Windows Registry Inventory examples:

- On Linux, collect metadata of .sh files in the /home/ec2-user directory, excluding all subdirectories. [{"Path":"/home/ec2-user","Pattern":["*.sh", "*.sh"],"Recursive":false}]
- On Windows, collect metadata of all ".exe" files in the Program Files folder, including subdirectories recursively. [{"Path":"C:\Program Files","Pattern":["*.exe"],"Recursive":true}]
- On Windows, collect metadata of specific log patterns. [{"Path":"C:\ProgramData\Amazon","Pattern":["*amazon *.log"],"Recursive":true}]
- Limit the directory count when performing recursive collection. [{"Path":"C:\Users","Pattern":["*.ps1"],"Recursive": true, "DirScanLimit": 1000}]

Use Systems Manager **Resource Data Sync** to send Inventory data collected from all of your managed instances to a single S3 bucket. Resource Data Sync then automatically updates the centralized data when new Inventory data is collected. With all Inventory data stored in a target S3 bucket, you can use services like Athena and QuickSight to query and analyze the aggregated data.

Inventory **integrates with Athena** to help you query inventory data from multiple AWS Regions and accounts. **Athena integration uses Resource Data Sync** so that you can view inventory data from all of your managed instances on the Inventory Detail View page in the SSM console. This feature **uses AWS Glue to crawl the data** in your S3 bucket, and **Athena to query the data**.

AWS Systems Manager - State Manager

AWS Systems Manager State Manager is a secure and scalable configuration management service that automates the process of keeping your Amazon EC2 and hybrid infrastructure in a state that you define.

- Can also be used to ensure that your managed instance are configured with specific applications, such as anti-virus or malware applications
- Can be used to automate the process of updating the SSM Agent or other AWS packages or ensure that specific ports are closed or open.

It can be used to ensure that:

- Instances are patched with specific software updates.
- Bootstrap instances with specific software at start-up
- Download and update agents on a defined schedule, including SSM Agent
- Configure network settings
- Join instances to a Windows domain (Windows instances only)
- Patch instances with software updates throughout their lifecycle
- **Run scripts on Linux and Windows managed instances throughout their lifecycle**
  - Can set a a schedule for when these scripts are applied (every Tuesday at 3AM, or every Monday at 5pm for example).

State Manager – Use Cases

State Manager integrates with AWS CloudTrail to provide a record of all executions that you can audit, and Amazon CloudWatch Events to track state changes.

- You can also choose to store and view detailed command output in Amazon S3.

A State Manager association is a configuration that is assigned to your managed instances.

- State Manager uses SSM documents to create an association.
- The configuration defines the state that needs to be maintained on your instances.
- An association can specify that anti-virus software must be installed and running on your instances, or that certain ports must be closed.
- The association specifies **a schedule for when the configuration is reapplied (every Tuesday at 3AM, or every Monday at 5pm for example)**.
- The association also specifies actions to take when applying the configuration.
- Example: a state manager association for anti-virus software is run once a day.
  - If the software is not installed, then State Manager installs it.
  - If the software is installed, but the service is not running, then the association might instruct State Manager to start the service.

AWS Systems Manager - Session Manager

- Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances through an **interactive one-click browser-based shell or through the AWS CLI**.
  - Session Manager provides secure and auditable instance management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys.

- It makes it easy to comply with corporate policies that require controlled access to instances, strict security practices, and fully auditable logs with instance access details, while still providing end users with simple one-click access

- Can be used to establish secure connections to both Windows and Linux Instances

Session Manager integrates with the following AWS services to provide logging and auditing capabilities:

- AWS CloudTrail
- Amazon S3
- Amazon CloudWatch Logs
- Amazon CloudWatch Events and Amazon Simple Notification Service

Session Manager – Use case

- An on-call engineer in your IT department receives notification, at 2AM, of an issue that requires him to remotely connect to an instance to troubleshoot/fix.
  - Using the AWS Systems Manager console or the AWS CLI, he can starts a session connecting him to the instance, runs commands on the instance needed to complete the task, and then terminates the session.

How It works?

- When he sends that first command to start the session, the Session Manager service:
  - Authenticates his ID,
  - Verifies the permissions granted to him by an IAM policy,
  - Checks configuration settings (such as verifying allowed limits for the sessions), and
  - Sends a message to SSM Agent to open the two-way connection.

- After the connection is established and he types the next command, the command output from SSM Agent is uploaded to this communication channel and sent back to his local machine.

AWS Systems Manager - Patch Manager & Distributor

SSM Capabilities – Actions (cont.) – Patch Manager

Used to automate the process of scanning managed instances for missing patching, and then updated their **OS security related patches.** patching the SSM managed instances.

- For windows instance, it is used for OS Security patching only
- For Linux instances, it can be used to update any packets on the instance

It can be used to apply any missing patches to instances **individually, or to a large groups** of instances by using Amazon EC2 instance tags.

- You can patch fleets of Amazon EC2 instances or your on-premises servers and virtual machines (VMs) by operating system type.

Security patching can be automated on a regular basis by scheduling patching to run as a Systems Manager Maintenance Window task.

SSM Capabilities – Actions (cont.) - Distributor

Used to create and deploy/publish software packages to SSM managed instances.

Clients can package own software packages or find AWS-provided agent software packages.

- Ex. Use it to install AmazonCloudWatchAgent on managed instances

One package, many platforms

- One document can have attached ZIP files that are installed on different operating systems

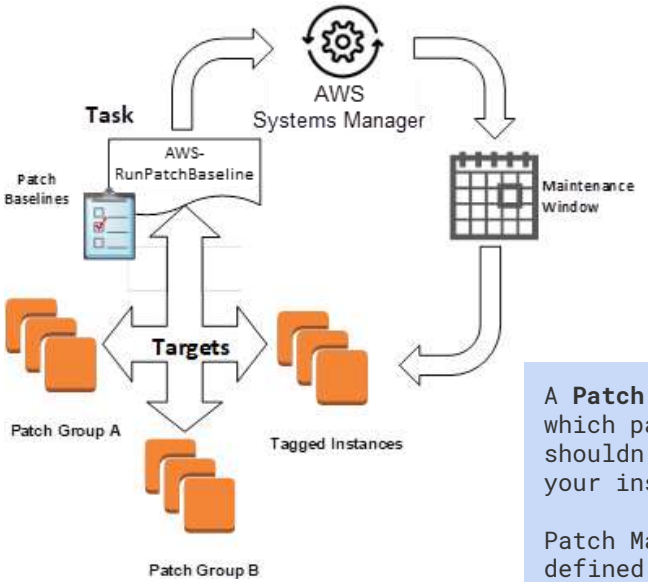
Control package access across groups of managed instances

- You can use **Run Command** or **State Manager** to control which of your managed instances get a package and which version of that package.
- Managed instances can be grouped by instance IDs, AWS account numbers, tags, or AWS Regions.

Automate deployment

- To keep your environment current, use **State Manager to schedule packages for automatic deployment** on target instances when those instances are first launched.

AWS Systems Manager - Example showing patching of windows instances



Step 1 → Use patch baselines to include rules for auto-approving patches within days of their release, as well to see a list of approved and rejected patches.

Step 2 → Leverage patch groups to organize instances for patching. E.g. create patch groups for dev/test/production.

Step 3 → Install patches on a regular basis by scheduling patching to run as a Maintenance Windows task

A **Patch baseline** defines which patches should and shouldn't be installed on your instances.

Patch Manager has a pre-defined patch baseline that approves all patches classified as critical updates or security updates with a severity of Critical or Important. These patches are automatically approved by this baseline seven days after they are released by Microsoft.

A **Patch group** is an optional means of defining which patch baseline should be used for what instances. E.g dev/test/production. You can also create primary and secondary failover cluster groupings .

Patch groups can be created based on server function, e.g. web servers and db. Patch groups can help you avoid deploying patches to the wrong set of instances.

SSM **Maintenance Windows** let you define a schedule for when to perform potentially disruptive actions on your instances such as patching an OS, updating drivers, or installing software. Each Maintenance Window has a schedule, a duration, a set of registered targets, and a set of registered tasks. Typically you want to apply your patches at a time when there is the least impact to your organization.

Parameter Store vs Secrets manager

	Parameter Store	Secrets Manager
Data can be stored as	Plain text & encrypted (w/ KMS)	Only encrypted (w/ KMS)
Integrate with IAM & CFormation	Yes	Yes
Stores value upto 4096 characters	Yes	Yes
Cost model	Free	Paid USD 40 cents / param & USD .05 cents / 10k Api calls
Secrets rotation	No	Automatic with RDS w/ lambda for other services
Generate random secrets	No	Yes
Cross account access	No	Yes

AWS Systems Manager - Use cases & Best practices

Automation
- Create self-service runbooks for infrastructure as Automation documents.
- Use Automation to simplify creating AMIs from the AWS Marketplace or custom AMIs, using public SSM documents or by authoring your own workflows.
- Build and maintain AMIs using the AWS-UpdateLinuxAmi and AWS-UpdateWindowsAmi Automation documents, or using custom Automation documents that you create.

Inventory
- Use Systems Manager Inventory with AWS Config to audit your application configurations over time.

Maintenance Windows
- Define a schedule to perform potentially disruptive actions on your instances such as OS patching, driver updates, or software installations.

Parameter Store
- Use Parameter Store to centrally manage global configuration settings.
- Use Parameter Store to encrypt and manage secrets by using AWS KMS.
- Use Parameter Store with ECS task definitions to store secrets.

Patch Manager
- Use patch manager to rollout patches at scale and increase fleet compliance visibility across your instances.

Run Command
- Manage Instances at Scale without SSH Access Using EC2 Run Command.
- Audit all API calls made by on or on behalf of Run Command using AWS CloudTrail.
- Use the targets and rate control features in Run Command to perform a staged command execution.
- Use fine-grained access permissions for Run Command (and all Systems Manager capabilities) by using IAM policies.

State Manager
- Update SSM Agent at least once a month using the pre-configured AWS-UpdateSSMAgent document.
- Bootstrap EC2 Instances on launch using EC2Config for Windows
- (Windows) Upload the PowerShell or DSC module to Amazon S3, and use AWS-InstallPowerShellModule.
- Use Amazon EC2 tags to create application groups for your instances. And then target instances using the Targets parameter instead of specifying individual instance IDs.
- Automatically remediate findings generated by Amazon Inspector by using Systems Manager.
- Use a centralized configuration repository for all of your SSM documents, and share documents across your organization.

Managed Instances
- Systems Manager requires accurate time references in order to perform its operations. If your instance's date and time are not set correctly, they may not match the signature date of your API requests. In some cases, this will lead to errors or incomplete functionality. For example, instances with incorrect time settings will not be included in your lists of managed instances.