Deployment Strategies - EC2

INPLACE upgrade vs DISPOSABLE upgrade

performing application updates on live Amazon EC2 instances. A disposable upgrade, on the other hand, involves rolling out a new set of EC2 instances by terminating older instances.

An in-place upgrade is typically useful in a rapid deployment with a consistent rollout schedule. It is designed for sessionless applications. You can still use the in-place upgrade method for stateful applications by implementing a rolling deployment schedule.

In contrast, disposable upgrades offer a simpler way to know if your application has unknown dependencies. The underlying EC2 instance usage is considered temporary or ephemeral in nature for the period of deployment until the current release is active. During the new release, a new set of EC2 instances are rolled out by terminating older instances. This type of upgrade technique is more common in an immutable infrastructure

Reference:

https://d1.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf#page=13

Tutorials Dojo's AWS Certified Solutions Architect Professional Exam Study Guide:

https://tutorialsdojo.com/aws-cheat-sheet-aws-certified-solutions-architect-professional/

Deployment Strategies - Elastic Beanstack

Deploying a new version of your application to an environment is typically a fairly quick process. The new source bundle is deployed to an instance and extracted. Then the web container or application server picks up the new version and, if necessary, restarts. During deployment, your application might still become unavailable to users for a few seconds. You can prevent this by configuring your environment to use rolling deployments to deploy the new version to instances in batches.

All at once – Deploy the new version to all instances simultaneously. All instances in your environment are out of service for a short time while the deployment occurs. If the deployment fails, a system downtime will occur.

Rolling – Deploy the new version in batches. Each batch is taken out of service during the deployment phase, reducing your environment's capacity by the number of instances in a batch. If the deployment fails, a single batch will be out of service.

Rolling with additional batch – Deploy the new version in batches, but first launch a new batch of instances to ensure full capacity during the deployment process. This is quite similar with Rolling option. If the first batch fails, the impact would be minimal.

Immutable – Deploy the new version to a fresh group of instances by performing an immutable update. If the deployment fails, the impact is minimal.

Blue/green deployment – Deploy the new version to a separate environment, and then swap CNAMEs of the two environments to redirect traffic to the new version instantly. If the deployment fails, the impact is minimal.

As shown above, the correct answer is ***all at once*** as that deployment option may result to a downtime. Hence, you should avoid using this type of option on your deployment activities. The following options provide a minimal impact in case that the deployment fails:

1. Rolling

2. Rolling with additional batch

3. Immutable

4. Blue/green

Reference:

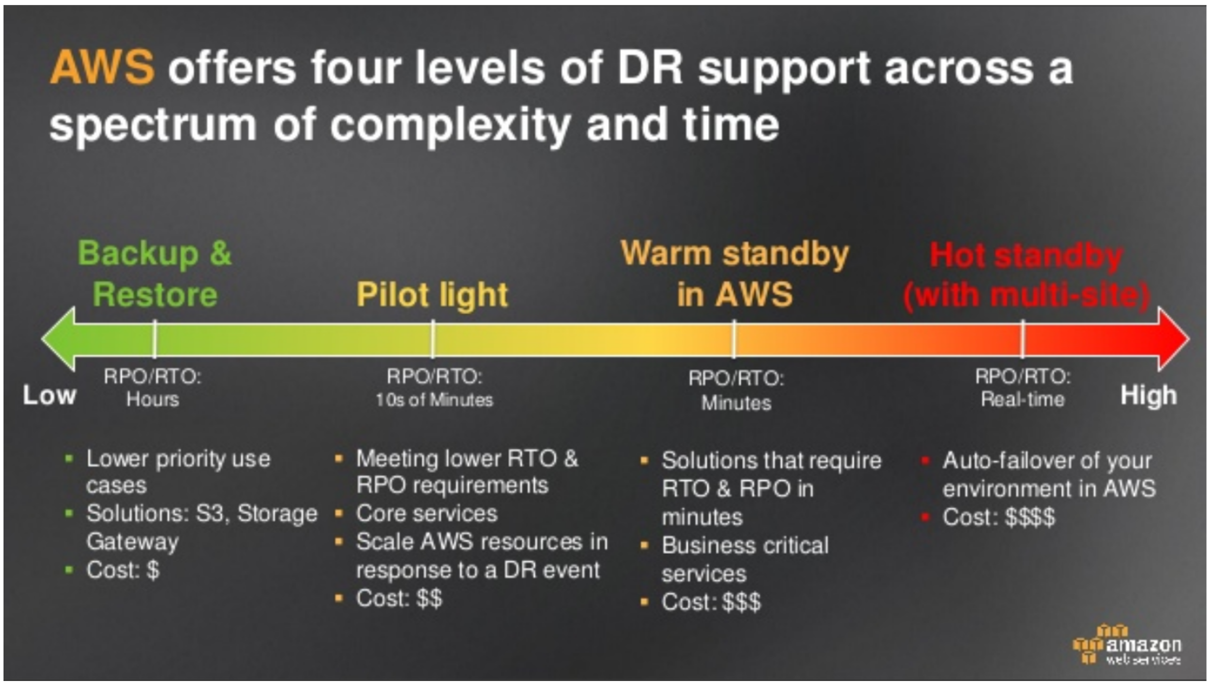
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html#deployments-newversion

Check out this AWS Elastic Beanstalk Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-elastic-beanstalk/

Disaster - Recovery

The term pilot light is often used to describe a DR scenario in which a minimal version of an environment is always running in the cloud. The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small flame that's always on can quickly ignite the entire furnace to heat up a house.



AWS offers four levels of DR support across a spectrum of complexity and time

Low — High

Backup & Restore
RPO/RTO: Hours
- Lower priority use cases
- Solutions: S3, Storage Gateway
- Cost: $

Pilot light
RPO/RTO: 10s of Minutes
- Meeting lower RTO & RPO requirements
- Core services
- Scale AWS resources in response to a DR event
- Cost: $$

Warm standby in AWS
RPO/RTO: Minutes
- Solutions that require RTO & RPO in minutes
- Business critical services
- Cost: $$$

Hot standby (with multi-site)
RPO/RTO: Real-time
- Auto-failover of your environment in AWS
- Cost: $$$$

amazon web services

This scenario is similar to a backup-and-restore scenario. For example, with AWS, you can maintain a pilot light by configuring and running the most critical core elements of your system in AWS. When the time comes for recovery, you can rapidly provision a full-scale production environment around the critical core.

Infrastructure elements for the pilot light itself typically include your database servers, which would replicate data to Amazon EC2 or Amazon RDS. Depending on the system, there might be other critical data outside of the database that needs to be replicated to AWS. This is the critical core of the system (the pilot light) around which all other infrastructure pieces in AWS (the rest of the furnace) can quickly be provisioned to restore the complete system.

References:

https://www.slideshare.net/AmazonWebServices/disaster-recovery-options-with-aws

https://d1.awsstatic.com/BackupRecovery/APN-Storage_Disaster-Recovery.pdf

Backup and Restore vs Pilot Light vs Warm Standby vs Multi-site:

https://tutorialsdojo.com/backup-and-restore-vs-pilot-light-vs-warm-standby-vs-multi-site/

Disaster - Resilient Infra

Requirements of a disaster-resilient network infrastructure

- Configure data replication to provide a durable storage.

- Run the accounting application both on-premises and in AWS with full capacity.
- Set up weighted DNS service in Route53 to route traffic across sites.

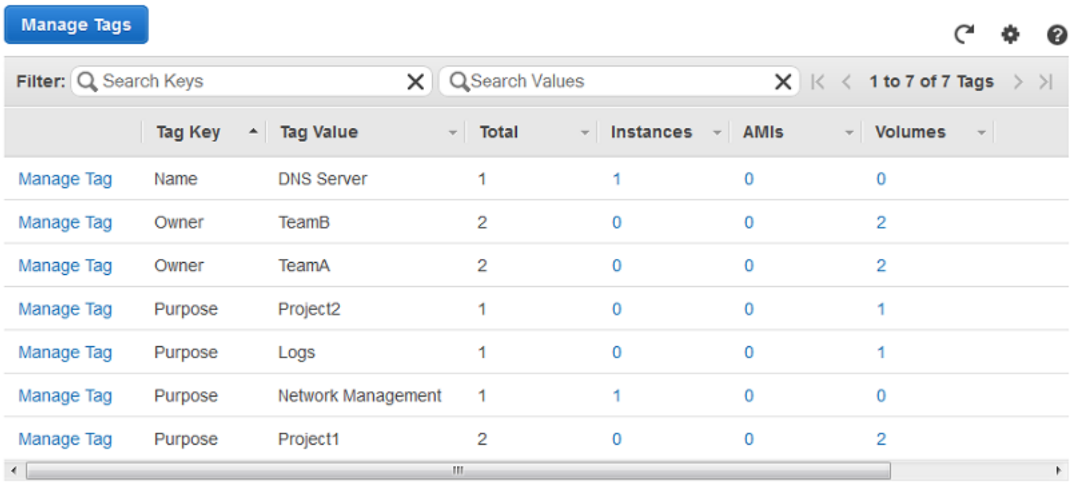
References:

https://www.slideshare.net/AmazonWebServices/disaster-recovery-options-with-aws

https://aws.amazon.com/disaster-recovery/

Governance - Manage Tags

AWS offers a variety of tools to help you implement proactive tag governance practices by ensuring that tags are consistently applied when resources are created.



Manage Tags

Filter: Search Keys | Search Values | 1 to 7 of 7 Tags

	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

AWS CloudFormation

- CloudFormation provides a common language for provisioning all the infrastructure resources in your cloud environment.
- CloudFormation templates are simple text files that create AWS resources in an automated and secure manner.
- When you create AWS resources using AWS CloudFormation templates, you can use the CloudFormation Resource Tags property to apply tags to certain resource types upon creation.

AWS Service Catalog

- Service Catalog allows organizations to create and manage catalogs of IT services that are approved for use on AWS.
- These IT services can include everything from virtual machine images, servers, software, and databases to complete multi-tier application environments.
- AWS Service Catalog enables a self-service capability for users, allowing them to provision the services they need while also helping you to maintain consistent governance – including the application of required tags and tag values.

AWS Identity and Access Management (IAM)

- IAM enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow or deny their access to AWS resources.
- When you create IAM policies, you can specify resource-level permissions, which include specific permissions for creating and deleting tags.
- In addition, you can include condition keys, such as **aws:RequestTag** and **aws:TagKeys**, which will prevent resources from being created if specific tags or tag values are not present.

References:

https://d1.awsstatic.com/whitepapers/aws-tagging-best-practices.pdf

https://aws.amazon.com/about-aws/whats-new/2018/03/aws-service-catalog-announces-autotags-for-automatic-tagging-of-provisioned-resources/