

Reference Architecture for a replicated ArcGIS Enterprise deployment on AWS

Last generated: October 22, 2026



Table of Contents

- **Reference Architecture**
 - Introduction and overview 2
 - Logical reference architecture 5
 - Implementation approach 10
 - Design considerations 16
 - Using this reference architecture 26

Introduction and overview

This reference architecture represents a replicated, geographically redundant deployment of ArcGIS Enterprise on Linux in Amazon Web Services (AWS). Its primary system is supported by a warm standby system in a second region and kept up to date using the [WebGIS DR tool](#). It is intended for organizations that operate ArcGIS Enterprise in the cloud and need a disaster recovery (DR) approach that limits downtime in the event of a regional failure or other catastrophic event.

- Learn [about reference architectures](#) first to gain important context.

Note:

This reference architecture represents an example blueprint for this system type. The design does not reflect the needs of all ArcGIS Enterprise deployments, and it is one of [several disaster recovery approaches](#) an organization might choose.

The design pairs two identical ArcGIS Enterprise deployments, one in each of two AWS regions. Only one of the two is reachable from the internet at any given time: DNS determines which site is active, and a scheduled backup and restore cycle keeps the standby site current. Recovery is performed by shifting DNS traffic to the standby site rather than by rebuilding a deployment from scratch.

Note:

This is a disaster recovery design, not a [high availability](#) design. The standby site only contains content up to the point of the last successful backup and restore cycle. See the [design considerations](#) for a fuller discussion of this tradeoff.

Capabilities

This architecture delivers disaster recovery capabilities for an ArcGIS Enterprise deployment, including:

- A warm standby deployment in a second, geographically separate AWS region, fully installed and configured rather than provisioned on demand after an outage or disaster event
- Identical configuration across both deployments, including the same web context URLs and domain name, so that clients and integrated systems do not need to change how they reach the system after a recovery

- A repeatable backup and restore cycle between the two sites using the WebGIS DR tool, carrying portal items, hosted content, and server configuration from the primary site to the standby site
- Automated, cross-region replication of backup artifacts so that a backup taken in the primary region is available in the standby region without manual transfer, complemented by a separate replication strategy for stores the backup does not cover, such as the S3 object store
- A DNS-driven recovery mechanism that shifts user traffic from the primary site to the standby site, with recovery time governed by DNS record time to live (TTL)
- A workflow that lends itself to scripting and automation, including backups, restores, traffic shifting, and detection of which site is currently active

The scenario used to develop this reference architecture places the primary site in a US East region and the standby site in a US West region, though the design applies to any pair of regions that meets an organization's data residency and latency requirements.

Software components

This reference architecture is based on ArcGIS Enterprise 11.5 on Linux, deployed to Amazon EC2 instances. The following ArcGIS components are included:

- [Portal for ArcGIS](#), the entry point for users and the machine from which WebGIS DR backups and restores are run.
- [ArcGIS Server](#) configured as a hosting server.
- [ArcGIS Data Store](#) configured as a relational store for persisting [ArcGIS-managed](#) data. In this design the object store is provided by Amazon S3 rather than by ArcGIS Data Store, as described below.
- An additional [ArcGIS Server](#) site configured with the GIS Server role, federated with the portal.
- The [WebGIS DR tool](#), a utility included with Portal for ArcGIS that creates a backup of a full ArcGIS Enterprise deployment and restores it to another deployment.

The design also depends on the following AWS services:

- **Application Load Balancer (ALB)** in each region, providing the HTTPS entry point to the deployment, supporting TLS with a certificate issued for the public domain, and routing requests to target groups for the portal, hosting server, and federated server.

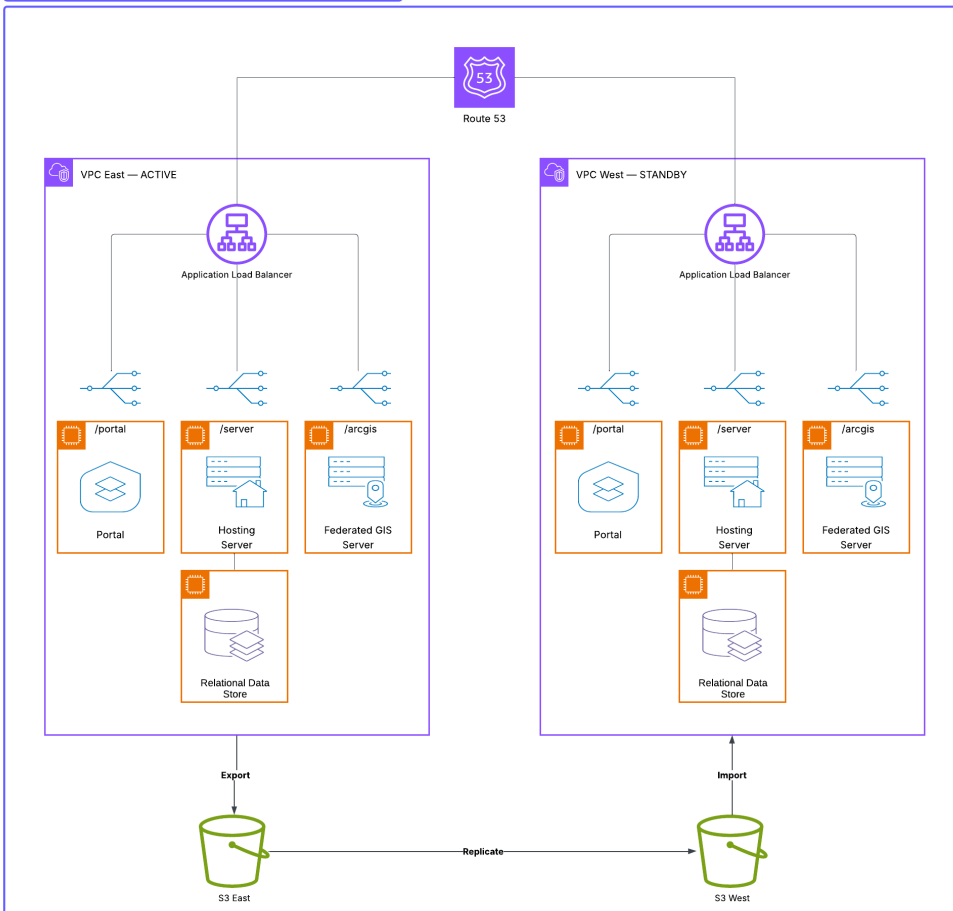
- **Amazon Route 53** for DNS, using a public hosted zone to control which region receives internet traffic and a private hosted zone in each region to allow both deployments to be configured with the same domain name.
- **Amazon S3** in three distinct roles, which are worth keeping separate when planning the design:
 - As the **ArcGIS Enterprise object store**. A folder in an S3 bucket is registered with the hosting server to act as the object store, in place of an ArcGIS Data Store object store. This is what allows members to publish hosted scene layers and 3D tile layers.
 - As the **WebGIS DR backup target**, using a bucket in each region with cross-region replication so that backups written in the primary region are copied to the standby region.
 - As part of the **ArcGIS Server configuration store**, when a **cloud-native configuration store** is used instead of a file share.
- **Amazon DynamoDB**, used together with S3 for the ArcGIS Server configuration store in a cloud-native storage configuration.
- **Amazon EC2** instances hosting the ArcGIS Enterprise components.

The role each component plays and the way they interact is described in the [logical reference architecture](#). A condensed summary of how the design is built and operated is provided in the [implementation approach](#).

Logical reference architecture

This logical view of the reference architecture defines the individual software components in each region, how they are separated or combined, and the key interactions between them.

AWS Route 53 — Active / Standby VPC Architecture



Learn more about [diagramming resources](#) for ArcGIS systems.

Note:

This reference architecture was designed with a focus on specific disaster recovery [capabilities](#). It is a logical view and does not present a level of detail that can be directly implemented. See the [implementation approach](#) for a summary of how the design is built and operated.

Software components

The design consists of two structurally identical ArcGIS Enterprise deployments, one acting as the primary and one acting as the warm standby, each in a separate AWS region. In the scenario used here, the primary deployment is in a US-East region and the standby site is in a US-West region.

Each region contains the same set of components:

- A [base ArcGIS Enterprise deployment](#), built on Amazon EC2 instances, including:
 - [Portal for ArcGIS](#), which serves as the entry point for users and as the machine from which WebGIS DR backups and restores are run.
 - [ArcGIS Server](#) configured as a hosting server.
 - [ArcGIS Data Store](#) configured as a relational store for persisting [ArcGIS-managed](#) data.
- An additional [ArcGIS Server](#) site configured with the GIS Server role and federated with the portal, illustrating how additional server roles participate in the replicated design.
- An **Application Load Balancer (ALB)** providing the HTTPS entry point for the region. The ALB terminates TLS using a certificate issued for the organization's public domain and routes requests to a target group for each ArcGIS component, so [load balancing and reverse proxy](#) responsibilities sit in front of the ArcGIS tier.
- **Amazon S3 storage** serving three separate purposes within the region. Treating these as distinct stores matters, because they are protected by different mechanisms:
 - A folder in an S3 bucket registered with the hosting server as the [ArcGIS Enterprise object store](#), in place of an ArcGIS Data Store object store. This holds live layer data such as hosted scene layers and 3D tile layers.
 - A bucket used as the destination for WebGIS DR backups.
 - Optionally, storage for the ArcGIS Server [configuration store](#), used together with **Amazon DynamoDB**, when a cloud-native configuration store is preferred over a file share.

Two components span both regions and are what make the pair of deployments behave as a single logical system:

- **Amazon Route 53** provides DNS for the system through two kinds of hosted zones:
 - A single **public hosted zone** that determines which region receives traffic from the internet. This zone controls how traffic is routed from clients to one of the two backend

systems, and is where a DNS update takes place to shift traffic to the correct backend system.

- A **private hosted zone in each region**, using the same domain name as the public hosted zone and associated with that region's VPC. The private zones are what allow both deployments to be configured with an identical domain name and web context URL while remaining independently resolvable and operable inside their own VPC.
- **Cross-region replication between the two S3 buckets**, an [AWS feature](#) which copies backup artifacts written in the primary region to the bucket in the standby region without manual transfer.

The [WebGIS DR tool](#), included with Portal for ArcGIS, is the mechanism that extracts current deployment state from the primary site to a single backup, which can then be restored to the standby site. It produces a backup of the full ArcGIS Enterprise deployment, including portal items, hosted content, and server configuration, and restores that backup into the standby deployment.

Note:

This reference architecture is based on ArcGIS Enterprise 11.5, where the ArcGIS Server configuration store can be placed on Amazon S3 and DynamoDB, but the ArcGIS Server directories still require a file share, typically provided by an additional virtual machine in each region. At ArcGIS Enterprise 12.1, both the [configuration store and the server directories](#) can be cloud-native across Amazon S3, DynamoDB, and Amazon SQS, which removes the need for that file share machine. Organizations deploying at 12.1 or later should evaluate this option, as it simplifies the per-region footprint and removes a component that would otherwise need to be duplicated and kept symmetric across both sites.

An enterprise identity provider is recommended for a single sign-on (SSO) experience, though not strictly required. Because both deployments are configured identically, the same authentication configuration applies to each. Learn more about [ArcGIS authentication models and providers](#).

Key interactions

The components described above interact with each other in the following ways:

- **Client applications reach the active site over HTTPS through a single, stable domain name.** Requests resolve through the public hosted zone to the ALB in whichever region is currently designated as active, and the ALB routes each request to the target group for the portal, hosting server, or additional federated server based on the request path. Because the

domain name does not change during a recovery, clients and integrated business systems continue to use the same endpoint. Learn more about [DNS, naming, and URLs](#).

- **The standby site receives no user traffic while it is inactive, but it must remain reachable.** Because the public hosted zone is configured to point to the primary, clients are not directed to it, and it can be administered and restored into without being exposed to normal user traffic. This is what allows backups to be restored into it on an ongoing basis while the primary site continues to serve users. It is important to distinguish this from the standby being unreachable: the network path from clients to the standby region must already exist and be permitted, because a switchover changes only name resolution.
- **Clients need open network access to both regions.** Shifting DNS does not create connectivity. Routing, firewall rules, security groups, any VPN or AWS Direct Connect paths, and any forward proxy allowlists must permit client access to both regional deployments before a switchover is attempted. If users can reach only the primary region, DNS will resolve to the standby after a failover and connections will still fail. The ArcGIS Enterprise [disaster recovery and replication guidance](#) makes the same point: wherever the standby environment is placed, confirm that ArcGIS Enterprise clients can access it when it is needed.
- **DNS resolution differs inside and outside a VPC, which is central to the design.** When a request originates from an EC2 instance in a VPC associated with a private hosted zone, the resolver first evaluates whether the private hosted zone's name matches the domain in the request. If there is no match, the request is forwarded to a public DNS resolver. If there is a match, the private hosted zone is searched for a matching record, and if none is found the resolver returns a non-existent domain (NXDOMAIN) rather than falling back to the public resolver. The practical consequence is that a private hosted zone with an overlapping namespace takes full responsibility for resolving every name under that domain inside its VPC.
- **ArcGIS components within a region communicate with each other as they would in any base deployment.** ArcGIS Server maintains persistent connections to ArcGIS Data Store, the federated GIS Server site communicates with the portal for identity and sharing, and each component is registered with its own ALB target group. Additional detail on interactions and ports is available in the ArcGIS Enterprise [product documentation](#).
- **Deployment state flows one direction at a time, from the active site to the standby site.** A WebGIS DR backup is created on the primary site's portal machine and written to the S3 bucket in that same region. Cross-region replication then copies the backup to the bucket in the

standby region, where it is restored on a regular schedule into the standby deployment. The standby site therefore reflects the state of the primary site as of the last successfully completed backup and restore cycle.

- **Recovery is a DNS operation rather than a rebuild.** Shifting a DNS record in the public hosted zone redirects users to the standby site. Because the standby deployment is already installed, configured, and populated, the time to recover is governed primarily by DNS propagation and the record's time to live (TTL) rather than by provisioning work. See [design considerations](#) for the reliability implications of this approach.

Note:

Because a private hosted zone with an overlapping namespace authoritatively answers for its entire domain within the associated VPC, records must also be added for any other applications in that VPC that rely on names under the same domain. Otherwise those applications will fail to resolve.

Implementation approach

The [logical reference architecture](#) describes what the system is composed of. This page summarizes how it is built and operated, in the order the work is typically performed. It is a condensed overview intended to convey the shape of the effort and the decisions involved, not a step-by-step procedure. **Note:**

Authoritative, step-by-step instructions are maintained in the product documentation and in the source material this reference architecture is based on. For the foundational backup and restore workflow, see [disaster recovery and replication](#) and the blog post [Migrate to a new machine in ArcGIS Enterprise using the WebGIS DR tool](#). See the ArcGIS Enterprise on AWS documentation for deployment guidance specific to your organization's chosen deployment method.

Before deployment

The DNS and load balancing foundation is established first, because both ArcGIS Enterprise deployments must be configured with the same domain name from the outset. Retrofitting this after the software is installed is considerably more difficult.

1. **Create an Application Load Balancer in each region.** Attach the appropriate TLS certificate for the public domain registered in Route 53 so that each ALB can terminate TLS. A target group is required when creating a load balancer, so create the portal target group at this point. It does not need registered targets yet.
2. **Create the public hosted zone records in Route 53,** using a CNAME type pointed at each regional ALB. The recommended approach is a **failover routing policy**: configure a primary record for the primary region's ALB, a secondary record for the standby region's ALB, and a Route 53 health check on the primary. Route 53 answers with the primary record while its health check passes and automatically switches to the secondary record when it fails.
 - Set the time to live (TTL) to **60 seconds** on both records. A short TTL limits how long clients already holding a cached answer continue to be sent to a failed or former endpoint after a switchover.
 - A manual single-record swap, where one record is edited to point at the standby ALB when a failure is confirmed, is a simpler alternative if automatic switchover is not needed. It cannot be automated, since there is no second record for a health check to fail over to.

- Weighted routing, with one record at weight 100 and the other at 0, is also workable as an on/off switch and can be paired with health checks. AWS positions weighted routing for proportional load balancing rather than active-passive failover, so the failover routing policy is the more direct fit here. If weighted routing is used, the weights must stay at 100/0; an intermediate split such as 70/30 would send live traffic to both regions, and the standby is not continuously consistent with the primary.
3. **Create a private hosted zone in each region**, using the **same domain name** as the public hosted zone, and associate it with the VPC that the region's ALB belongs to. Depending on the organization's network configuration, existing hosted zones or regional zones may be appropriate. Consult with your organization's IT department and DNS management to understand options and choose the best approach.
 4. **Add a matching record to each private hosted zone.** The record should mirror the one in the public hosted zone. The default TTL of 300 seconds and simple routing policy are appropriate here.

The last two steps are what make the rest of the design work. Two private hosted zones sharing a domain name with the public zone allow both regions to host an identically configured ArcGIS Enterprise deployment, and allow the standby deployment to stay operational inside its own VPC so that restores can be performed against it while the primary site serves users.

This is also where the approach differs most from the equivalent on-premises workflow. On-premises migrations commonly use `etc/hosts` entries on each machine to resolve the shared public URL to a local component. Here, the per-region private hosted zones provide that name resolution at the VPC level instead, so no host file entries are needed on the EC2 instances.

Note:

A private hosted zone with an overlapping namespace answers authoritatively for that entire domain within its associated VPC. If other applications in the same VPC depend on names under that domain, add records for them to the private hosted zone as well, or they will stop resolving.

ArcGIS Enterprise deployment

The following is performed identically in both regions, producing two deployments that differ only in which one receives public traffic.

1. **Deploy the base ArcGIS Enterprise deployment software** (Portal for ArcGIS, hosting server, and ArcGIS Data Store) to EC2 instances. Assign the instances appropriate VPCs and security groups to enable ingress from that region's ALB.
2. **Deploy the additional GIS Server site** using the same approach. It will be federated with the portal in a later step, once the shared DNS alias is in place.
3. **Create the remaining target groups** for the hosting server and the federated server. The portal target group already exists from the ALB creation step. Depending on the system's logical architecture, one target group may be sufficient.
4. **Register each instance with its corresponding target group**, then **configure the ALB forwarding rules** to route requests to the correct target group. At this point the portal should be reachable from within the VPC through the DNS name defined in the internal Route 53 records.
5. **Set the portal's system properties** to the shared DNS alias rather than to a machine name, so that the deployment identifies itself by the same name in both regions. This means setting `privatePortalURL` and `WebContextURL` to the portal URL built on the shared domain name.
6. **Federate the hosting server and the additional GIS Server** with the portal. The administrative URL used during federation may differ depending on the level of administrative access configured for the web tier.

When this is complete, both regions hold a fully functional deployment. The one currently answered by the public hosted zone's primary record is reachable from the internet and acts as the primary site. The other receives no public traffic while it is inactive and acts as the warm standby, though it remains reachable on the network so that it can be administered and restored into.

Note:

The WebGIS DR tool requires certain settings to be identical across the two deployments, and the specifics vary by ArcGIS Enterprise version. The portal's public URL and the service URLs used to federate ArcGIS Server sites are the most significant. Review the compatibility requirements in the [disaster recovery and replication documentation](#) for the version you are deploying before running a restore.

Configure backup and replication storage

Backups need to exist in both regions without an operator moving files between them.

1. **Create an S3 bucket in each region**, using names that make the region and purpose obvious, such as a `-backup-east` and `-backup-west` convention.
2. **Enable versioning on both buckets**. This is a prerequisite for cross-region replication.
3. **Configure replication on the primary region's bucket**, selecting the standby region's bucket as the destination. The default settings are generally appropriate.

S3 Replication Time Control is available as an option and replicates 99.99% of new objects within 15 minutes. It is worth considering for large deployments with substantial content and tight downtime expectations.

Replicate the object store separately

The WebGIS DR backup does **not** include the object store when the object store is a folder in an S3 bucket. The backup and the object store are two different stores, and only one of them is handled by the tool. The object store therefore needs its own replication strategy.

1. **Replicate the object store bucket to the standby region independently**. A scripted copy using the AWS CLI, run on a schedule, keeps the standby region's object store aligned with the primary. This is separate from, and in addition to, the replication configured on the backup bucket.
2. **Point the WebGIS DR properties file at the backup bucket**, not at the object store bucket. With `BACKUP_STORE_PROVIDER = AmazonS3`, the backup bucket and region are named by the `S3_BUCKET` and `S3_REGION` properties, with `PORTAL_BACKUP_S3_BUCKET` used for portal content. Keeping the backup bucket and the object store bucket clearly distinguished in configuration avoids a situation where the backup appears to succeed while hosted scene layers and 3D tile layers are missing after a restore.

Because the object store and the WebGIS DR backup are replicated by different mechanisms on potentially different schedules, the two can drift apart. Coordinating their timing keeps the standby site internally consistent.

Note:

This design restores from a bucket in the standby region, which keeps the restore local. If you instead restore from a bucket in a different region than the target portal, confirm that no VPC endpoint is configured that would prevent the portal from making the outbound requests needed to restore content. This is a straightforward thing to overlook in a cross-region design and it surfaces only at restore time.

Disaster recovery and failover

With both deployments in place and buckets replicating, the ongoing operational cycle is a backup on the primary site and a restore on the standby site.

1. **Create a backup of the primary site** using the WebGIS DR tool, run on the machine where Portal for ArcGIS is installed. Point the properties file at the S3 bucket in the same region as the primary site. Cross-region replication copies the backup to the standby region automatically. The actual step to initiate the backup can be scripted and scheduled using operating system tooling. Consider incremental backups for larger systems to increase speed of availability.
2. **Restore the replicated backup on the standby site**, again running the tool on the standby site's portal machine. Reference the standby region's local bucket in the restore properties file rather than the primary region's bucket.
3. **Fail over.** If a failover routing policy was configured, this happens automatically once the health check on the primary fails. Point the health check at a meaningful ArcGIS endpoint rather than at the load balancer alone, and set failure thresholds so that a brief network problem does not trigger a switchover. If switchover is manual, an administrator edits the single record or toggles the weighted records once a failure is confirmed. The resulting downtime depends on how quickly the switch happens, by health check or by operator, plus the TTL on the records and, for a partial rather than total outage, how long clients continue reaching the impaired primary before their cached answer expires. See [cross-region switchover at the DNS level](#) for the full tradeoffs between these approaches.

How often the backup and restore cycle runs determines how much data could be lost in a disaster, so the schedule is effectively a recovery point objective decision. See [design considerations](#) for more on this tradeoff.

If the original primary site is unavailable for an extended period, the direction of the backup and restore cycle needs to be reconsidered so that content created on the newly active site is preserved when the original region is brought back online.

Because the entire cycle consists of tool invocations and API-driven DNS changes, it lends itself to scripting. Automating the backups, restores, the DNS switchover itself where a failover routing policy isn't already handling that automatically, and the detection of which site is currently active, reduces both the operational burden and the risk of error during an actual disaster.

Design considerations

The design needs to meet certain criteria within each [architecture pillar](#). Because this reference architecture exists to serve a disaster recovery objective, reliability considerations dominate and are presented first. The practices outlined below do not represent a complete set of architectural considerations.

You can also refer to these [physical design considerations](#) for additional recommendations.

Reliability

[Reliability](#) ensures your system provides the level of service required by the business, as well as your customers and stakeholders. This architecture addresses one specific reliability concern: the loss of an entire region or data center. It does so by maintaining a second, fully configured deployment that can be made active without a rebuild.

Recovery objectives

The two measures that should drive the configuration of this design are the recovery point objective (RPO), meaning how much data the organization can afford to lose, and the recovery time objective (RTO), meaning how quickly service must be restored.

- **RPO is determined by the backup and restore schedule.** The standby site reflects the state of the primary site as of the last successfully completed WebGIS DR backup and restore cycle. Anything created or edited after that point is not present on the standby. A daily cycle implies up to a day of potential data loss; a more frequent cycle narrows the gap at the cost of additional processing and storage.
- **RTO is largely determined by DNS and by how quickly the switchover is triggered.** Because the standby deployment is already installed, configured, and populated, recovery does not include provisioning or installation time. If a person has to notice the failure and act, that detection delay counts toward the RTO. Once the switchover happens, residual delay from DNS caching may or may not matter depending on the failure type. A full regional outage forces clients onward regardless, while a partial failure or planned failover depends more on the time to live (TTL) on the Route 53 records. See [cross-region switchover at the DNS level](#) for the full

discussion. Also note that during the restore process, a failure and switchover may lead to an unexpected application state, and this should be tested fully.

Note:

This design is a disaster recovery approach and is **not** a [highly available](#) ArcGIS Enterprise deployment. Failover can be fast, but the standby site only contains content up to the timestamp of the last successful backup and restore. Organizations that require continuous availability with no data loss should review [configuring highly available ArcGIS Enterprise](#), and can combine high availability within each region with the cross-region replication described here.

Backups

A replicated standby site does not remove the need for a conventional [backup](#) strategy. The WebGIS DR backup is the mechanism that carries state between sites, but it also serves as a restore point for scenarios that are not regional failures, such as an administrative error or content corruption. Because replication propagates the primary site's state to the standby, a problem introduced on the primary site will also reach the standby on the next cycle. Retaining multiple backup generations, rather than only the most recent, protects against this.

Enabling versioning on the S3 buckets is a prerequisite for cross-region replication in this design, and it also contributes to backup durability by preserving prior object versions.

What the backup does not cover

Treating the WebGIS DR backup as a complete copy of the system is the most consequential mistake in this design. Several things fall outside it and need their own protection, or the standby site will be incomplete in ways that only become visible during a recovery.

- **The object store.** When the object store is a folder in an S3 bucket, as in this design, it is not included in the WebGIS DR backup. It holds live layer data such as hosted scene layers and 3D tile layers, and it requires a separate replication strategy, typically a scheduled copy to the standby region's bucket using the AWS CLI. The WebGIS DR properties file should reference the backup bucket, while the object store bucket is replicated independently.
- **Other exclusions documented for the tool.** At ArcGIS Enterprise 11.5, map and tile service caches, referenced data sources for web services, and spatiotemporal big data store backup files are [not included in a WebGISDR backup](#) and must be restored separately. Referenced data sources are worth particular attention, since an enterprise geodatabase registered with the site is the responsibility of its own database replication or backup strategy, not the WebGIS DR tool.

- **Constraints on the restore target.** The organization being restored to must be at the same ArcGIS Enterprise version as when the backup was created, and on the same operating system family. This makes upgrades a coordinated, two-region activity rather than something done to one site at a time.

The practical implication is that the recovery point for the system as a whole is set by the *least* current of these mechanisms, not by the WebGIS DR schedule alone. If the backup runs hourly but the object store is copied nightly, the effective recovery point for hosted scene layers is nightly. Aligning the schedules, or at least understanding where they differ, is what makes the stated RPO real.

Note:

Because these stores are replicated by different mechanisms, they can drift out of step with one another. A restore that completes successfully can still produce a standby site where portal items reference layer data that has not yet arrived. Coordinating the timing of the WebGIS DR cycle and the object store copy, and validating the standby after a restore, is what prevents this.

Symmetry between sites

The WebGIS DR tool depends on the two deployments being equivalent, and this constrains the design more than any other factor:

- The number of machines and the ArcGIS Enterprise component layout should match across the two sites.
- The URLs used to reach the portal and the service URLs used to federate ArcGIS Server sites must be the same in both sites. This is the reason the design uses a shared domain name backed by per-region private hosted zones rather than region-specific names.
- ArcGIS Server directory names, and the names of folders registered with the server sites, must match between sites. The underlying paths may differ, but the names cannot.
- Requirements vary by ArcGIS Enterprise version. Confirm the specifics for the version being deployed before relying on a restore.

Any configuration change made to the primary site should be treated as a change that must also be reflected at the standby site, either through the restore cycle or as a deliberate parallel change.

Replication timing

Cross-region replication introduces a delay between a backup being written in the primary region and being available in the standby region. For most deployments this delay is small relative to the

backup schedule itself. Where downtime expectations are tight or backup artifacts are large, **S3 Replication Time Control** provides a predictable bound by replicating 99.99% of new objects within 15 minutes, at additional cost. Sizing the backup and measuring actual replication time is a better basis for this decision than assuming either outcome.

Network routing

A switchover changes name resolution and nothing else. It does not create network connectivity, and this is a common way for an otherwise sound disaster recovery design to fail at the moment it is needed.

Users and integrated systems must have open network access to **both** regional deployments before a switchover is attempted. That means routing, firewall rules, security groups, any VPN or AWS Direct Connect paths, and any forward proxy allowlists all need to permit traffic to each region. If connectivity has only ever been proven against the primary region, a failover will move DNS successfully and still leave users unable to connect.

This is worth verifying explicitly rather than assuming, particularly where clients reach the system from an internal corporate network rather than the public internet, and particularly for any client population that sits behind a different network path than the one used for testing. Refer to [secure network design](#) when planning how both regions are reached, and note that this requirement coexists with keeping the standby site appropriately protected.

Cross-region switchover at the DNS level

Because DNS decides which region serves users, the switchover mechanism is a design decision in its own right rather than an operational afterthought. Route 53 and the ALB operate at different layers here: Route 53 decides *which region* a client is sent to, while the ALB distributes requests *within* whichever region it belongs to. The ALB has no awareness of the other region, so the cross-region switchover is entirely a Route 53 concern.

Route 53 supports several routing policies, and they are not interchangeable for this design. [AWS documentation on active-active and active-passive failover](#) is explicit that active-passive failover is configured using the **failover routing policy**, while any other routing policy, including weighted, is used for active-active configurations where all returned resources are expected to serve traffic. This design is active-passive: exactly one region should ever be live. That makes the failover routing policy the recommended mechanism, with two alternatives worth knowing about.

Failover routing policy (recommended). A primary and secondary record are configured explicitly, together with a Route 53 health check on the primary. Route 53 answers with the primary record as long as the health check passes, and automatically answers with the secondary record when it fails. This expresses the active-passive intent directly rather than approximating it, and it is the only option described here that supports automatic switchover without being repurposed from a different use case.

Manual single-record swap. A single record is pointed at the primary ALB, and an administrator edits it to point at the standby ALB when a failure is confirmed. This is the simplest possible setup and a reasonable choice if switchover will always be a manual, deliberate action. It cannot be automated, because there is no second record for a health check to fail over to.

Weighted routing, used strictly as an on/off switch. AWS positions [weighted routing](#) for proportional load distribution and canary testing, not for active-passive disaster recovery. It can be made to work here by setting one record's weight to 100 and the other's to 0, which routes all traffic to the primary and none to the standby, and health checks can be attached so an unhealthy record stops receiving traffic. This behaves like a cruder version of the failover routing policy rather than offering any advantage over it, so it is included here mainly because it is a common pattern, not because it is the preferred one.

Note:

Weighted records must only ever be used at 100/0 in this design, never at an intermediate split such as 70/30 or 50/50. An intermediate weight sends live traffic to both regions at once, and the standby is not continuously consistent with the primary: it only reflects the state as of the last backup and restore cycle. Splitting traffic would mean two sets of users working against two diverging systems, with edits made on the standby being overwritten by the next restore. Proportional splits are only safe when both targets share a consistent data tier, which an active-passive design like this one does not have.

Regardless of which policy is used, two things govern how quickly clients actually follow the switch:

- **Detection, if a human is involved.** An operator-initiated change, whether editing a single record or toggling weights, is only as fast as the time it takes someone to notice the failure and act. Staffing and alerting, not the DNS mechanism itself, are usually the limiting factor here.
- **DNS caching, governed by TTL.** After Route 53 starts answering with the new record, any client or resolver holding a cached answer continues using it until that cache expires, which the TTL bounds. A short TTL, such as the 60 seconds used in this design, limits how long that can persist. How much this matters depends on the kind of failure:

- **In a full regional outage, the unresponsive primary does most of the work.** A client with a stale answer tries to connect, gets nothing, and the resulting connection failure is what drives re-resolution. The TTL is a secondary factor here, since the failure itself is what moves clients along.
- **In a partial or “gray” failure, TTL matters much more.** If the primary is degraded rather than fully down, for example slow or intermittently erroring, it may still accept connections from clients with a stale answer, who continue to be served by the impaired region until their cache expires.
- **In a planned or precautionary failover, the primary is healthy and will happily serve stale-routed clients for the full TTL.** This is the case where a short TTL earns its keep most directly.
- Health-check automation removes detection latency, not DNS caching latency. Clients still wait out the TTL after Route 53 changes its answer, with or without automation.

Whichever approach is chosen, the switchover path should be exercised rather than assumed, including a check of how it behaves under a partial failure, not only a full outage.

Failover and failback

Failover is triggered by a change in which record Route 53 answers with, whether that is an automatic response to a failed health check under a failover routing policy, a manual edit to a single record, or a manual toggle of weighted records. An automatic approach removes detection latency; a manual approach keeps confirmation of a genuine failure in human hands. The choice should follow from the recovery time objective and the organization’s operational model.

Failback deserves explicit planning. If the standby site serves as the active site for an extended period, content created there will not exist in the original region, and simply resuming the original backup direction would discard it. The direction of the backup and restore cycle needs to be reversed while the standby is active, and a deliberate cutover plan is needed to return to the original region. Testing this path is as important as testing the initial failover.

Testing

A recovery capability that has not been exercised should not be relied upon. Because the standby site is reachable within its own VPC, it can be validated in place after each restore without affecting production traffic. Periodically performing a full failover, ideally as a scheduled exercise, is the only way to confirm that both the technical steps and the operational procedures work.

Security

[Security](#) protects your systems and information. Replicating a deployment across regions widens the system's footprint, so the security posture of both sites needs to be equivalent.

- **TLS termination and certificates.** HTTPS is terminated at the Application Load Balancer in each region using a certificate for the shared public domain. Both regions need a valid certificate, and certificate renewal has to be handled in both places or a failover will surface an expired certificate.
- **Exposure of the standby site.** While its public weight is zero, the standby deployment receives no user traffic even though it remains fully functional. It should be treated as a production system for patching, hardening, and access control purposes regardless, since it is one DNS change away from serving all users. Note the balance to strike here: the standby must not be casually exposed, but it also cannot be locked away, because clients need a working network path to it before a switchover. See [network routing](#).
- **Identity provider configuration.** If the organization uses SAML or OpenID Connect, specific preparation is required before restoring a backup to the standby site so that members can still sign in after a failover. In the SAML case, both environments must be configured with the same SAML certificate. Confirming this before a disaster, rather than during one, is important. Learn more about [ArcGIS authentication models and providers](#).
- **Backup contents and access control.** WebGIS DR backups contain the organization's content and configuration. The S3 buckets holding them should be restricted, encrypted, and covered by the same access policies as the deployments themselves. Replication extends that requirement to the destination bucket and region.
- **Network design.** Each region's VPC, subnet, and security group design should follow [secure network design](#) practices, and the two regions should be consistent so that a failover does not change the system's effective security posture.

Observability

[Observability](#) provides visibility into the system, enabling operations staff to keep it running in a healthy, steady state. This design introduces observability requirements that a single-site deployment does not have.

- **Both sites need monitoring.** A standby site that has silently stopped accepting restores provides no recovery capability. Monitoring the standby with the same rigor as the primary is

what makes it trustworthy. See [monitoring system performance](#) and [ArcGIS and IT infrastructure monitoring](#).

- **The backup and restore cycle needs to be observable.** Success or failure of each backup, the replication of the artifact to the standby region, and the outcome of each restore should all be reported. A failed cycle directly degrades the achievable RPO, so it warrants alerting rather than passive logging.
- **The active site must be unambiguous.** Operations staff and automation both need a reliable way to determine which region is currently serving traffic. Deriving this from the DNS configuration, rather than from institutional memory, avoids errors during an incident.
- **Telemetry should be comparable across regions.** Capturing the same [telemetry](#) in both regions makes it possible to tell whether the standby site can actually carry production load, and to detect drift between the two.

Automation

[Automation](#) reduces effort spent on manual deployment and operational tasks, and reduces human-introduced errors. This architecture benefits from automation more than most, because its operational cycle is repetitive and its recovery path is executed under pressure.

- The backup, replication, and restore cycle can be scheduled and run without operator involvement.
- Traffic shifting is an API-driven DNS change, so failover itself can be scripted.
- Detection of which site is active can be derived programmatically and used to drive the other automation safely.
- Building both regions with infrastructure as code, and configuring ArcGIS Enterprise through automation, is the most dependable way to keep the two sites symmetric over time. Manual, parallel configuration of two environments tends to drift, and drift is what causes restores to fail.

Integration

[Integration](#) connects this system with other systems across the enterprise. The significant integration property of this design is that it is largely transparent to consumers.

Because both deployments are configured identically and are reached through the same domain name, services consumed by other business systems continue to resolve to the same URLs after a

failover. Integrated applications generally do not need to be reconfigured or made aware of which region is active.

Two caveats are worth noting. Integrations that reach the deployment by machine name or region-specific endpoint, rather than through the shared domain name, will not follow a failover and should be identified and corrected. Integrations that push data into the system on a schedule may need to account for the possibility that recent writes were not captured in the last backup before the failover.

Performance and scalability

[Performance and scalability](#) considerations for each individual site are the same as for any comparable single-region ArcGIS Enterprise deployment, and are not restated here. Two points are specific to this design:

- **The standby site should be sized to carry production load.** A standby provisioned with smaller instances to reduce cost will become the production system at the moment of a failover. If the sites are deliberately asymmetric in capacity, that decision should be explicit and its consequences understood.
- **Client latency may change after a failover.** Users and integrated systems that were close to the primary region may be considerably further from the standby region. This is an inherent consequence of geographic separation and worth setting expectations around, since the same separation is what provides protection.

Cost

Maintaining a warm standby means paying for a substantially complete second deployment that serves no user traffic under normal conditions, plus storage and replication costs for backups in two regions. This is the central tradeoff of the design. Organizations should weigh it against the business cost of extended downtime, and against alternatives such as reprovisioning the second site from automation at the time of a disaster, which reduces steady-state cost in exchange for a longer RTO.

Physical design considerations

The considerations here are primarily focused on logical architecture design, which must eventually be transformed into a physical architecture design. Esri offers [system architecture design](#) services should you need help determining the factors relating to your organization's physical design, such as networking, storage, system environments, and sizing.

Additional resources related to physical design include:

- [High availability](#)
- [Backups and disaster recovery](#)
- [Load balancing and reverse proxies](#)
- [Secure network design](#)
- [DNS, naming, and URLs](#)
- [Cloud services and providers](#)
- [Storage considerations](#)
- Minimum system requirements, listed in [ArcGIS product documentation](#).

Using this reference architecture

This reference architecture is presented as part of the ArcGIS Architecture Center to help organizations understand how they might combine ArcGIS and cloud infrastructure components in a standardized and well-defined configuration for disaster recovery. To make best use of it, the following approach is suggested:

1. **Assess** whether a warm standby approach matches your recovery objectives. Compare the [recovery point and recovery time implications](#) of this design against your organization's requirements, and confirm that a recovery point tied to a scheduled backup cycle is acceptable. If you require continuous availability with no data loss, [high availability](#) within a region is the more appropriate starting point, and can be combined with this approach.
2. **Use** the [logical reference architecture](#) as a starting point for your own design. Where your requirements differ, such as a different pair of regions, additional federated server roles, or a highly available configuration within each region, the [design process](#) can provide useful considerations to resolve those differences.
3. **Review** the [implementation approach](#) to understand the sequence of work and, in particular, the decisions that are difficult to change later. Establishing DNS and the shared domain name before installing software is the most consequential of these.
4. **Incorporate** the [design choices and considerations](#) with respect to your organization's non-functional requirements in areas like [reliability](#), [security](#), [observability](#), and [automation](#).
5. **Plan to test** the recovery path, not just build it. Schedule failover exercises and validate the standby site after restores. A disaster recovery design that has never been exercised carries unknown risk.
6. **Learn** more about the [architecture practices](#) illustrated in this reference architecture.

Esri offers [system architecture design](#) services should you need help in the design process.

Additional related resources

These resources reference key concepts and practices that may not have been explicitly stated within the reference architecture, but can be significant contributors to successfully designing, implementing, and operating a replicated ArcGIS Enterprise deployment.

- [Backups and disaster recovery](#)
- [High availability](#)
- [Reduce downtime](#)
- [Load balancing and reverse proxies](#)
- [Secure network design](#)
- [DNS, naming, and URLs](#)
- [Cloud services and providers](#)
- [Healthy system maintenance](#)
- [Disaster recovery and replication](#) in the ArcGIS Enterprise documentation