



Edge-to-Cloud Resiliency & Disaster Recovery



(800) 266-7798
www.inductiveautomation.com

Edge-to-Cloud Resiliency & Disaster Recovery

Resiliency in industrial automation is essential to minimizing downtime, preventing data loss, and ensuring safety. Achieving resiliency requires systems that can withstand, recover from, and adapt to disruptions through high uptime, fault tolerance, and effective disaster recovery strategies.

Ignition supports resilient architectures with features such as built-in redundancy and failover, store-and-forward to maintain data integrity, and support for edge computing and MQTT to enable decentralized processing and reduce reliance on central servers. The Enterprise Administration Module (EAM) further enhances resiliency by providing centralized management and automated backup capabilities.

While Ignition offers strong native tools for resiliency, integrating external cloud IT solutions, such as those available from AWS, further strengthens disaster recovery through scalable backup, failover, and replication strategies that extend beyond Ignition's built-in capabilities.

Understanding Resiliency in SCADA Systems

Resiliency in SCADA systems is crucial to ensure continuous operations despite failures or disruptions. Key principles like high availability, failover, and redundancy are central to achieving this goal. High availability ensures minimal downtime for critical processes, while failover enables automatic switching to backup systems during failures. Redundancy eliminates single points of failure by duplicating essential components, ensuring uninterrupted operations.

Traditional SCADA systems often rely on hardware-based solutions for redundancy and failover, which can be costly, inflexible, and challenging to scale.

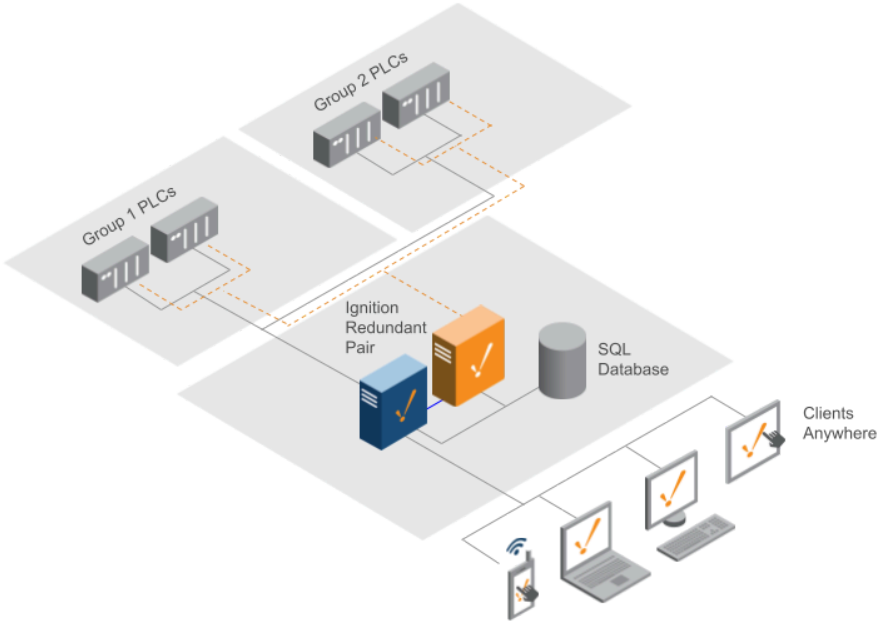
In contrast, Ignition takes a modern, software-driven approach that maximizes flexibility and scalability. It provides cold- and warm-standby redundancy for gateways, allowing backup systems to take over seamlessly without specialized hardware. Ignition's ability to integrate with cloud and edge computing enables distributed, scalable architectures that maintain high availability and data integrity across locations. The system's distributed design ensures real-time synchronization and failover, supporting continuous operations even in complex environments. Additionally, Ignition is operating-system-agnostic, allowing it to run seamlessly on any platform, further enhancing its flexibility.

Built on open IT standards, Ignition is a highly adaptable solution that integrates with existing infrastructure, offering a more cost-effective and future-proof approach to SCADA resiliency.

Ignition’s Built-in Resiliency Features

Ignition is designed to deliver high availability, fault tolerance, and overall resiliency. The platform’s built-in features, such as redundancy, failover, and integration with cloud and edge computing, make it a robust solution for industrial operations. Below are key features that support Ignition's resiliency capabilities:

Redundancy & Failover



Ignition’s master-backup architecture ensures that local, critical system functions continue uninterrupted, even in the event of a failure. In this setup, the master system handles all active processes, while the backup system mirrors its state, remaining in sync. Should the master system fail, the backup takes over automatically without the need for manual intervention.

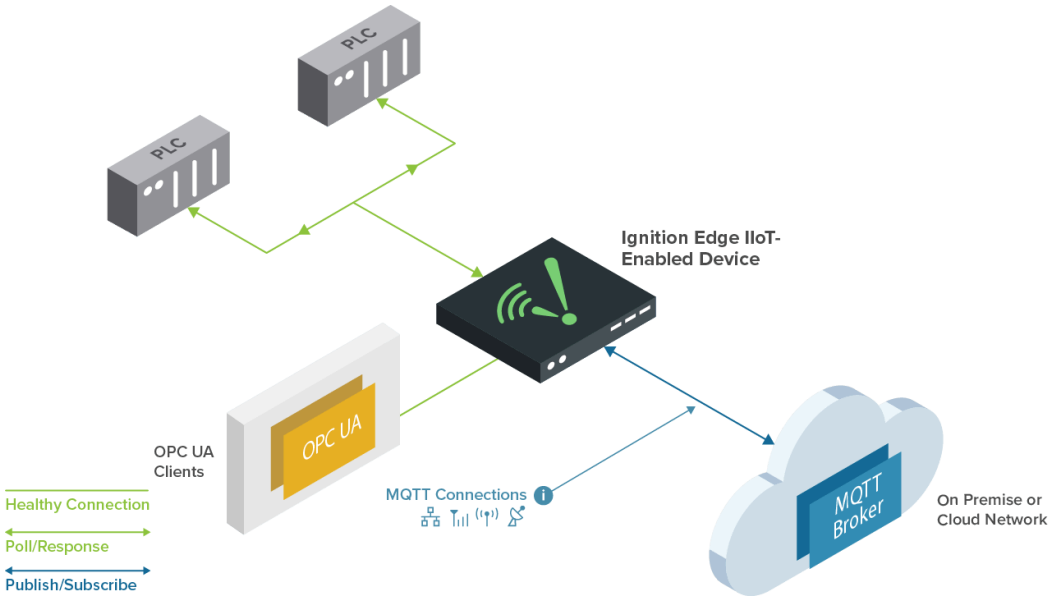
Automatic failover is enabled through seamless synchronization of data and processes between the primary and backup systems. When a failover occurs, Ignition ensures that no data is lost by synchronizing all Gateway configurations, tags, projects, etc., between the systems, allowing for smooth transitions with minimal impact on operations.

Store-and-Forward Mechanism

The store-and-forward mechanism ensures data integrity during network interruptions by temporarily buffering the data when the system loses connectivity and then forwarding it once the connection is restored. This feature helps maintain the continuity of data flow without gaps, even if communication between systems is disrupted.

For optimal performance, it's essential to configure this feature properly. Best practices include setting appropriate data buffer sizes, ensuring time-stamped data is synchronized, and regularly monitoring the communication channels to minimize data loss.

Edge Computing & MQTT

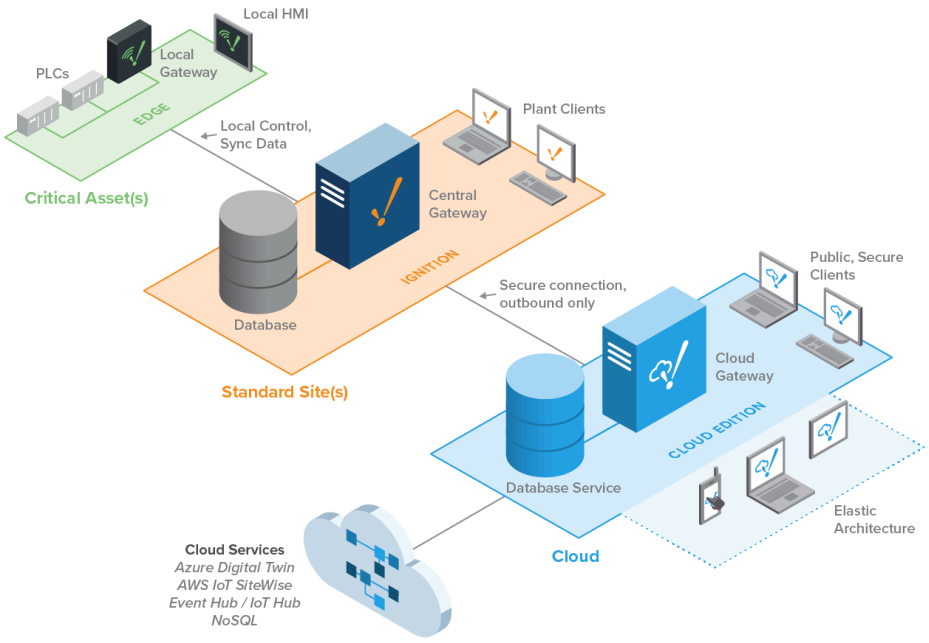


Ignition Edge products provide enhanced resiliency by decentralizing data processing to the edge. This reduces dependency on central servers, improving data availability and ensuring that local operations can continue even if the connection to the main system is temporarily lost. Edge computing also minimizes latency, making it easier to process data in real time, closer to its source.

In addition to its real-time data processing, the Edge Gateway also features store-and-forward capabilities, similar to those found in standard Ignition gateways. This ensures that historical data is safely stored locally during network interruptions and can be forwarded to the central system once connectivity is restored. This helps prevent data loss and guarantees the integrity of historical data.

When combined with MQTT Sparkplug, Ignition ensures reliable data transmission even in environments with intermittent connectivity. MQTT's lightweight, publish-subscribe protocol allows for efficient and secure data transmission, making it ideal for mission-critical systems where data availability and integrity are paramount.

Cloud & Hybrid Resiliency Strategies



Ignition's flexibility extends to cloud environments, where it can be deployed to enhance resiliency and scalability. Cloud-based implementations allow for centralized management, distributed failover, and increased redundancy without relying on on-premise hardware.

Additionally, hybrid architectures, combining on-premise and cloud-based systems, provide the best of both worlds. This strategy leverages cloud computing's scalability and cost-effectiveness while maintaining local control and data processing capabilities, ensuring high availability and resilience across diverse network environments.

Designing a Resilient Ignition Architecture

Designing a resilient Ignition architecture involves careful planning to ensure that the system can withstand disruptions and continue to operate effectively. Key considerations include distributed architectures, data flow continuity, version control, and utilizing Ignition's built-in tools like the Enterprise Administration Module (EAM) to streamline operations.

Distributed Architectures

Ignition can be architected in a multitude of ways, customized to meet the specific requirements of the operation and infrastructure. This adaptability allows it to conform to a wide range of environments, from small, localized systems to large, enterprise-scale solutions with geographically distributed components.

Ignition's modular design enables the architecture to grow and scale seamlessly. Whether adding more gateways, integrating new devices, or expanding to the cloud, Ignition can adjust easily without causing downtime or requiring a full system overhaul.

Load-Balanced Perspective Frontend

Application load balancing refers to the process of distributing incoming network traffic across multiple servers or instances to ensure that no single server is overwhelmed. For Ignition, load balancing helps manage and optimize the distribution of user requests and application workloads across multiple Ignition Gateways, ensuring that each Gateway handles an appropriate amount of traffic. This improves system performance, scalability, and availability by ensuring even distribution of workloads and preventing server overload. In environments with high user or application demands, load balancing can enhance responsiveness, reduce latency, and prevent bottlenecks that might otherwise occur with a single Gateway.

- Auto-Scaling: Load-balanced architectures can leverage auto-scaling to dynamically add or remove Gateway instances based on demand. This ensures that sufficient resources are available during peak usage periods while reducing costs during low-demand times. Ignition Cloud Edition fits this model very well, as it is a pay-as-you-go pricing model, meaning you only pay for the Gateway when it is up and running.
- Set Number of Applications: In some environments, a fixed number of Gateway instances is preferred for security and predictability. This approach ensures tighter control over resources and minimizes variability in system behavior, making it ideal for regulated industries.
- Security Considerations: Load balancing enhances security by distributing application traffic, reducing the risk of overloading any single Gateway. Additionally, load balancers can include features like SSL termination and intrusion detection, providing an extra layer of protection for critical applications.
- Scaling Options: Scaling can be achieved vertically by increasing the resources of individual Gateways or horizontally by adding more Gateways to the network. Horizontal scaling, combined with Ignition's Gateway Network and Cirrus Link's MQTT Modules, offers flexibility for geographically distributed operations or environments with diverse application needs.

By leveraging load balancing alongside Ignition's Gateway Network and Cirrus Link's MQTT Modules, organizations can achieve a highly scalable and resilient architecture tailored to their operational and security requirements. This complements, rather than replaces, redundancy for backend reliability.

VM Hypervisor-based Redundancy

VM Hypervisor-based redundancy for Ignition environments offers a highly flexible and efficient way to ensure high availability and fault tolerance. By running Ignition Gateways on virtual machines (VMs) on the same or different physical servers, this approach leverages the benefits of virtualization to provide seamless failover in case of hardware or software failures. The hypervisor can quickly spin up the backup VM, ensuring minimal downtime and continuous operation. With features like dynamic resource allocation, VMs can be adjusted to handle varying system loads, and in case of resource constraints, the backup VM can be used to balance the load.

One of the key advantages of VM-based redundancy is the ability to utilize snapshots and cloning. Snapshots allow for easy backup and recovery of both primary and backup VMs, and VMs can be cloned to create identical configurations for rapid deployment. Additionally, virtualization enables geographic redundancy by distributing VMs across multiple data centers, ensuring that in the event of a localized failure, the backup can be brought online at another location. Virtualization platforms like VMware or Hyper-V further simplify the management of redundant VMs with centralized tools that provide monitoring, maintenance, and automation.

VM hypervisor-based redundancy is cost-effective because it eliminates the need for dedicated physical hardware and allows for flexible scaling of resources. As the system grows, VMs can be provisioned as needed to handle increased demand, providing a scalable solution without requiring significant infrastructure investment. Furthermore, centralized management and backup strategies ensure that VMs can be quickly restored or replicated in the event of a failure, supporting disaster recovery efforts.

Comparing Ignition Redundancy To VM Snapshots and Cloning

In an Ignition system, warm redundancy offers a faster failover process compared to VM-based failover configurations. This is because, in a warm redundant setup, the backup Ignition Gateway is actively running and continuously synchronized with the primary Gateway. It subscribes to tags and maintains communication with devices, so when failover occurs, the backup is already in an operational state, requiring only a few final steps to seamlessly take over. This prepped state significantly reduces the time required for failover, ensuring minimal downtime and a quicker recovery compared to the more complex VM failover process.

Ignition Containerization with Docker

Containers provide a lightweight, efficient approach to application deployment by packaging everything needed to run a service — including the application code, runtime, and dependencies — into a standalone unit. While sharing the host operating system's kernel, containers maintain isolation from each other and start up significantly faster than traditional VMs since they don't require a full operating system boot. Inductive Automation supports containerization through their official Ignition image, which can be deployed using Docker for single-host container management or scaled across multiple hosts using container orchestration tools like Kubernetes.

For simpler production systems, Docker provides basic resilience features through container restart policies and health monitoring. Containers can be configured to automatically restart after crashes or system reboots using Docker's built-in restart policies. Health checks monitor the container's status and can trigger restarts if the application becomes unresponsive. While these features provide basic redundancy for single-host deployments, they don't offer the advanced orchestration capabilities needed for larger, distributed systems. That's where container orchestration platforms like Kubernetes become essential, particularly for enterprise-scale deployments.

Kubernetes enhances container-based resilience through several key mechanisms. It continuously monitors container health and automatically restarts failed containers, while pod replication ensures multiple copies of your Ignition instances run across different nodes. Unlike VM-based redundancy, Kubernetes can rapidly reschedule pods (containers) to healthy nodes if a host fails, recovering quickly. The platform also enables zero-downtime updates through rolling deployments, gradually replacing old containers with new versions while maintaining availability. These orchestration capabilities complement other resilience methods, working alongside load balancers and autoscaling groups to distribute traffic and maintaining high availability even during maintenance windows or infrastructure failures.

Enterprise Administration Module (EAM) for Resiliency

The Enterprise Administration Module (EAM) enhances resiliency by providing centralized control over multiple Ignition gateways. It streamlines system management and backup automation, and enables proactive monitoring, reducing downtime and ensuring smooth operations.



Centralized System Management

- Monitor and configure multiple gateways from a single interface.
- Manage licensing, modules, and project deployments across distributed architectures.
- Deploy project updates efficiently without manual intervention.

Automated Backup & Recovery with EAM

- Schedule and distribute backups across gateways, ensuring data integrity.
- Automate project backups and deployments, reducing manual effort.

Remote Gateway Health Monitoring & Alerts

- Detect system failures before they impact operations.
- Set up automated alerts for proactive maintenance, improving system uptime.

By integrating EAM, organizations can simplify gateway management, improve failover response, and maintain a resilient SCADA environment with minimal manual intervention.

Additional Ignition Failover Mechanisms

In addition to its core resiliency features, Ignition offers several other capabilities that further enhance system reliability. Built-in IDP failover ensures continuous authentication availability by automatically switching to a local user source if the connection to the primary fails. Database failover configuration and the ability to connect to database clusters provide high availability and data integrity, minimizing the risk of data loss. The Tag History Splitter enhances reliability by distributing tag history across multiple databases, reducing single points of failure. Additionally, remote MQTT broker connectivity enables efficient, decentralized communication, further strengthening system uptime and fault tolerance.

Git Revision Control & Project Updates

Ignition projects are stored on the file system in a readable format, making Git integration straightforward. Using Git for version control ensures structured project management, allowing teams to track changes, collaborate efficiently, and roll back updates when needed.

With Git, projects can be versioned, branched, and merged, enabling safe testing before deployment. Combined with the Enterprise Administration Module (EAM), updates can be centrally managed and rolled out across multiple gateways, ensuring consistency and reducing deployment risks.

By leveraging Git and EAM, Ignition users can maintain a resilient, well-managed SCADA environment with minimal disruption.

Ignition SCADA Disaster Recovery On AWS

In the above section, we have seen how native redundancy at the edge (on-premise) is achieved with Ignition SCADA. While this is acceptable for most organizations, there are requirements where a full stack disaster recovery (DR) mechanism has to be implemented. Ignition SCADA consists of the following three components:

1. Frontend Perspective
2. Backend Ignition Gateway
3. Database

A full stack Ignition SCADA disaster recovery on AWS has to achieve the following:

1. Seamless failover to a secondary Ignition SCADA in another AWS availability zone or region, kept in sync with primary SCADA.
2. Redundant and secure networking for immediate connectivity between asset and secondary SCADA.

Let us see how such a SCADA disaster recovery solution can be achieved on AWS. Consider the following two scenarios.

Scenario 1 – Inter Availability Zone (AZ) Disaster Recovery

In this scenario, we deploy Ignition SCADA in a virtual private cloud (VPC) spread across two AWS availability zones (AZ). The Ignition stack is deployed in AZ-1. In our case, we use **AZ us-east-1a as the Primary SCADA AZ**. There is an identical stack deployed in AZ-2, **us-east-1b, which is the Secondary SCADA** and is ready to take over in the event that AZ-1 becomes unavailable.

Scenario 2 – Inter Region Disaster Recovery

In this scenario, we extend the DR mechanism to compensate for a complete region failure. This is a rare scenario, however in the event of a catastrophic region failure or debilitating cyber-attack, the disaster recovery infrastructure deployed in the second AWS region will immediately take over the SCADA operations. In our case, we use **North Virginia as the Primary SCADA region and Ohio as our DR or Secondary SCADA region**.

Now let's look in detail at a resilient AWS reference architecture for both of the above scenarios.

Scenario 1 – Inter AZ Disaster Recovery on AWS

Below is the reference architecture for this scenario.

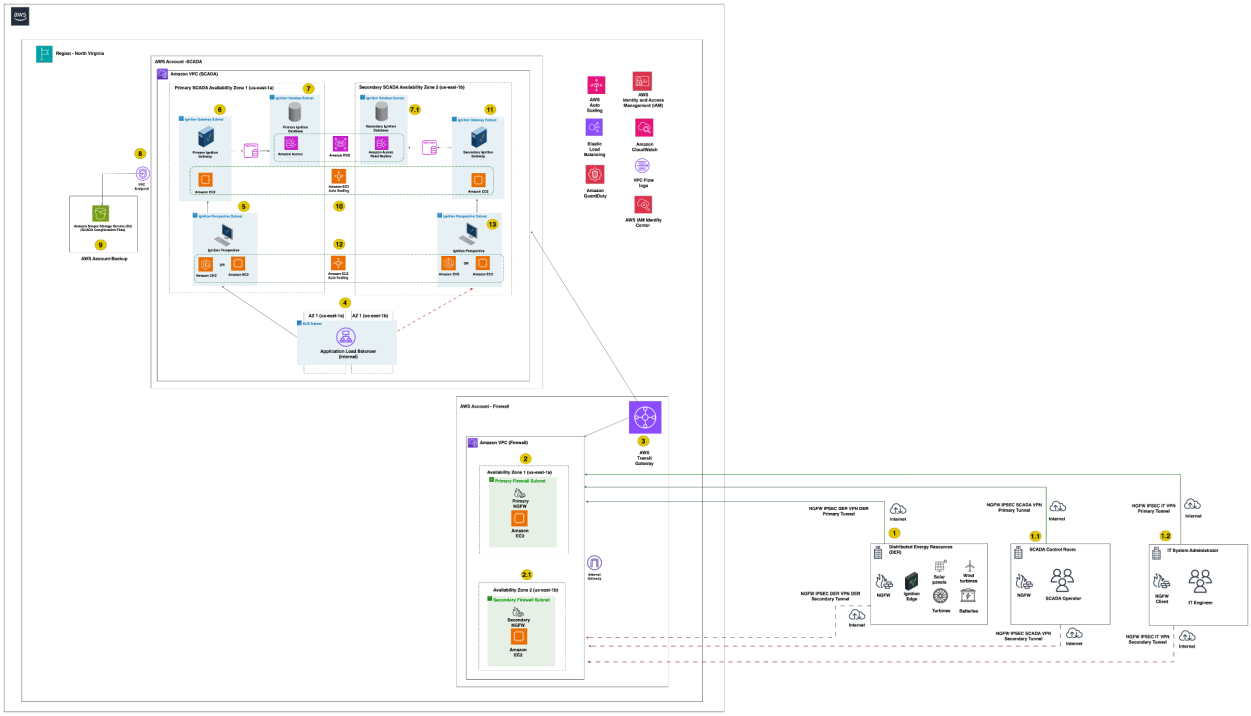


Figure 1 – Inter AZ Disaster Recovery

There are three types of resources at the edge that make a connection with the AWS cloud. The first type of resources are **distributed energy resources (DER)**. These resources can be solar panels, wind turbines, batteries or any other type of operations or generation technology. They can also be assets from manufacturing, automotive, pulp and paper, etc. AWS services are agnostic of industry, size, scale, capacity, or cost of the asset.

The DER site can also deploy Ignition Edge. Ignition Edge is a lightweight version of the Ignition platform, designed for edge computing applications. It allows users to collect, process, and analyze data from devices and sensors at the edge of a network, closer to where the data is generated. This can help reduce latency, improve response times, and decrease bandwidth usage by minimizing the amount of data sent to a central server or cloud.

The DER assets need to be connected over a secure and private VPN connection, to the Ignition SCADA running in the AWS cloud.

The second type of resources are the **SCADA operators** that need to access the HMI screens of the Ignition Perspective which is running in the cloud. The SCADA engineers need to have secured VPN access to the Ignition SCADA to carry out control and monitoring operations of the DER assets.

The third type of resources are the **IT Engineers.** They need to carry out maintenance work on the machine where Ignition Gateway is installed e.g., firmware and patch upgrades, installation of device drivers, etc.

All three resource categories make their own **independent IPSEC VPN** connections to the AWS cloud. The resources do not share the VPN connections with each other. Additionally, the access

levels for the OT assets, SCADA operators, and IT engineers are strictly regulated, adhering to the principles of **least privilege and task-specific authorization.**

SCADA users usually always have an incumbent security appliance of choice, e.g., Fortinet, Cisco, Palo Alto, etc. They prefer to use the native VPN functionality of the security appliance, due to familiarity with the product. Customers can also use AWS Site-to-Site VPN.

A VPN is needed to establish 100% private connectivity from edge to the cloud. If a dedicated **physical connection** from edge to cloud is preferred, AWS Direct Connect can be used.

For security reasons, we have **three separate AWS accounts each hosting a different section of the cloud infrastructure.**

- AWS SCADA account to host the Ignition stack
- AWS Backup account to store Ignition Gateway configuration files
- AWS Firewall account to host the Next Generation Firewall (NGFW)

Now, let's go over the architecture steps:

1. A generation site at the edge where distributed energy resources (DER) are located. The DER assets are secured behind a Next Generation Firewall (NGFW). The NGFW instance at the DER site, makes **two secure IPSEC VPN** connections to a cloud version of the NGFW hosted in the AWS cloud.

One VPN tunnel is the Primary and the other is Secondary. Both VPN tunnels are **always active**, however traffic is sent over only one tunnel. In a normal situation, the Primary tunnel is used. When DR occurs, traffic immediately gets diverted to the Secondary tunnel.

This failover mechanism can be configured in the NGFW console.

1.1. The SCADA Control Room, secured behind a Next Generation Firewall (NGFW). The NGFW instance at the SCADA Control Room makes **two secure IPSEC VPN** connections to a NGFW hosted in the AWS cloud. One VPN tunnel is the Primary and the other is Secondary. Both VPN tunnels are **always active**, however traffic is sent over only one tunnel. In a normal situation, the Primary tunnel is used. When DR occurs, traffic immediately gets diverted to the Secondary tunnel.

This failover mechanism can be configured in the NGFW console.

1.2. The device of the IT Engineer e.g., a laptop, tablet etc. on which a client of the Next Generation Firewall (NGFW) is installed. The NGFW client also has a choice of **two secure IPSEC VPN** connections. One Primary and one Secondary. The client determines which tunnel to use based on AZ availability.

2. An Amazon Virtual Private Cloud (VPC) called **Firewall VPC.** It is this VPC, where we deploy the cloud version of the security appliance, i.e., NGFW on Amazon EC2. We deploy two instances of the NGFW: one in AZ-1 us-east-1a to function as the Primary and the other in AZ-2 us-east-1b to function as the Secondary in case of DR. They can be maintained in **Active-Active or Active-Passive** mode. The NGFW are in their separate AWS **Firewall account.**

The NGFW instance in AZ-1, us-east-1a, is the Primary FW and terminates the Primary VPN tunnels from 1, 1.1 and 1.2. When a VPN connection is successfully established, the route tables associated with the subnet divert the traffic to the AWS Transit Gateway.

2.1. The NGFW instance in AZ-2, us-east-1b, is the Secondary FW and terminates the Secondary VPN tunnels from 1, 1.1 and 1.2. When a VPN connection is successfully established, the route tables associated with the subnet divert the traffic to the AWS Transit Gateway. This Secondary FW is only used when the Primary FW is lost due to AZ failure.

3. AWS Transit Gateway or TGW connects your Amazon VPCs and on-premises networks through a central hub. This connection simplifies your network and puts an end to complex peering relationships. Transit Gateway acts as a highly scalable cloud router — each new connection is made only once.

In our case, the TGW is connected to the Firewall VPC and the SCADA VPC. As more VPCs are added based on requirement, they can all connect to the TGW, from where the TGW route table can determine the movement of the traffic.

In our case, the TGW acts as a hop for the NGFW to forward the traffic to the Application Load Balancer or ALB in the SCADA VPC.

4. An internal-facing ALB is hosted in two AZs: us-east-1a and us-east-1b. The ALB is part of the SCADA VPC. The SCADA VPC is in its separate AWS Account, called a **SCADA account**.

The SCADA VPC is where the DR mechanism is hosted. It is in this VPC that the Ignition stack is deployed. The VPC is spread across two AZs: us-east-1a and us-east-1b. The Primary Ignition SCADA is deployed in us-east-1a and the Secondary or Disaster Recovery SCADA is deployed in us-east-1b. In a normal scenario, the Primary SCADA is available in us-east-1a, hence the ALB diverts all traffic to targets in us-east-1a. In the event when AZ-1 us-east-1a becomes unavailable, the DR mechanism will launch new infrastructure in DR AZ us-east-1b (Steps 10 to 12). At that moment, the ALB will start sending traffic to us-east-1b.

5. Ignition Perspective frontend is hosted on an Amazon EC2 instance or on Amazon EKS. Ignition Frontend Gateways [TG3] can access the Ignition Backend [TG4] Gateway, which is hosted in a different subnet, but is part of the same VPC. All subnets in a VPC can natively communicate with each other; however, we can control traffic based on source, destination ,and protocol. For this, we use an AWS service called Security Groups.

6. The Ignition Backend [TG5] Gateway is hosted on Amazon EC2. It connects to the Ignition database via an Amazon Relational Database Service (Amazon RDS) Proxy. An Amazon Relational Database Service (Amazon RDS) Proxy is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. The RDS proxy ensures that the Ignition Gateway always has an immediate active port to connect to.

Also, the primary Ignition SCADA has a **scheduled backup** configured to store its Gateway Backup file to an Amazon S3 bucket. This frequency to backup can be as low as one minute.

7. The Ignition Database, hosted on Amazon Aurora PostgresSQL. Amazon Aurora PostgreSQL is a fully managed, PostgreSQL-compatible, and ACID-compliant relational database engine that combines

the speed, reliability, and manageability of Amazon Aurora with the simplicity and cost-effectiveness of open-source databases.

7.1. A read replica of the Amazon Aurora PostgresSQL is maintained in the DR AZ us-east-1b. Read replicas allow you to create one or more copies of your Aurora database that can be used to offload read traffic from the primary database instance. By having a read replica of the Primary Ignition database, we ensure that a copy of the database is **always ready and available in DR AZ us-east-1b**. In the event of AZ-1 (us-east-1a) failure, the Aurora read replica in AZ-2 (us-east-1b) is automatically promoted as the **writer node.**

8. Using VPC Endpoints, a backup of the Primary Ignition configuration is saved in an Amazon S3 bucket using AWS private backbone.

9. As part of security best practices, the Amazon S3 bucket is in a separate AWS Account, called **Backup account.**

We enable the S3 Object lock feature. S3 Object Lock prevents data from being altered or deleted by any person or process, whether unintended or because of malicious activity. This helps prevent ransomware events from deleting or altering your data. To further protect against cyber threats, we can enable S3 encryption. Amazon S3 provides several encryption options to secure your data both at rest and in transit.

AZ Disaster Recovery Mechanism

AZ disaster recovery is achieved as follows:

2.1. The NGFW present at resources 1, 1.1 and 1.2 identifies that the Primary VPN tunnels have collapsed due to failure to reach the endpoint of Primary NGFW in us-east-1a. At that moment, the Secondary tunnels take over to send traffic to Secondary NGFW in DR AZ us-east-1b. The secondary NGFW in DR AZ us-east-1b becomes active and begins to receive traffic from the three resources 1, 1.1 and 1.2.

10. Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. Here we have an Auto Scaling Group or ASG which has been configured with a policy that dictates the availability of at least one and maximum one EC2 instance of the Ignition Frontend Gateway. We configure the policy to use both AZ-1 us-east-1a and AZ-2 us-east-1b. In a regular scenario, the ASG will launch a new EC2 in AZ-1 us-east-1a. However, when AZ-1 us-east-1a becomes unavailable, it will automatically spin up a new EC2 in AZ-2 us-east-1b, based on a preselected launch template. It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances.

Once such parameter is a User Data Script. When we launch an Amazon EC2 instance, we can pass user data to the instance that is used to perform automated configuration tasks after the instance starts.

In this script, we add instructions to read the latest version of the Primary Ignition Gateway backed-up configuration file in **Amazon S3 from step 9** above.

11. The user data script, as part of the ASG launch template, spins up the Secondary Ignition Gateway, based on the latest version of the Primary Ignition Gateway configuration file.

Furthermore, the RDS proxy is always connected to the already available read replica of the Primary Ignition database. The RDS proxy ensures the availability of an active database connection that can be immediately used by the launched Secondary Ignition Gateway.

12. Similar to step 10, here we have an ASG, configured to launch an Amazon EC2 instance or Amazon EKS. We configure the policy to use both AZ-1 us-east-1a and AZ-2 us-east-1b. In a regular scenario, the ASG will launch a new EC2/EKS in AZ-1 us-east-1a. However, when AZ-1 us-east-1a becomes unavailable, it will automatically spin up a new EC2/EKS, in AZ-2 us-east-1b, based on a preselected launch template.

13. The Perspective HMI projects are originally saved on the Ignition Frontend Gateway, which has just been spun up in step 11. So, when Ignition Perspective EC2/EKS become available, they connect with the Secondary Ignition Gateway. At that point, all the HMI screens are immediately available.

One important component[TG6] of the Ignition SCADA is the database. Using the read replica feature of Amazon Aurora in step 7.1, we have a version of the database, which is **available and in sync** with the Primary Ignition database, **24x7, 365 days a year.**

Using the above setup, a full-scale AZ disaster recovery can be achieved in a few minutes.

Scenario 2 – Inter Region Disaster Recovery on AWS

For this scenario, we'll consider the reference architecture below.

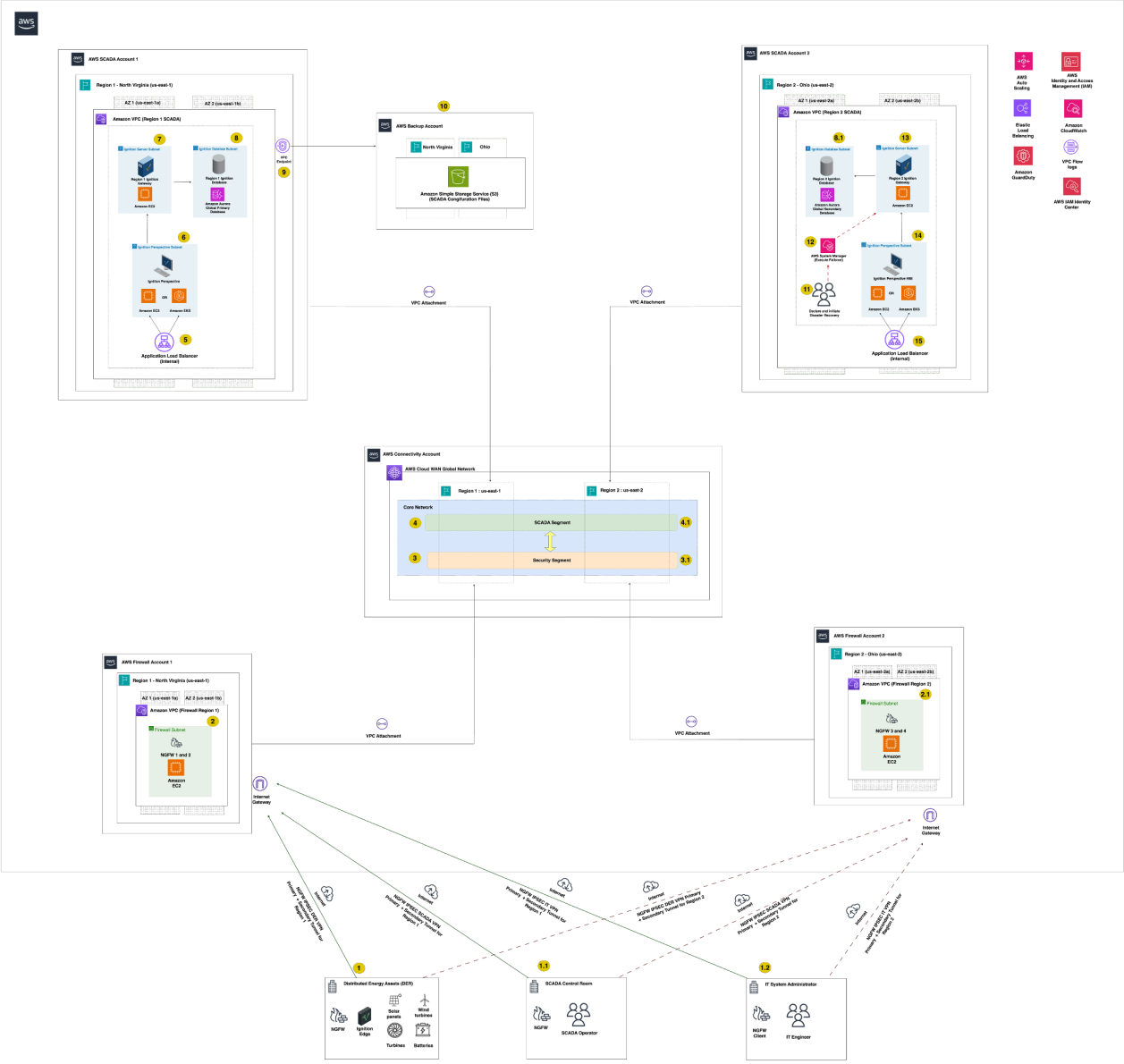


Figure 2 – Inter Region Disaster Recovery

This solution already has the inter AZ disaster recovery built in, as we saw in the previous section. We now raise the bar and prepare for a catastrophic region failure or cyber-attack. It is extremely rare that an AWS region will become unavailable. However, for compliance, safety, and security concerns, it is possible that customers want to have a fully functioning Ignition SCADA in a different region altogether.

Similar to AZ DR methodology, we start by segregating and isolating the infrastructure based on roles and responsibilities. This is because if there was just one account for the entire infrastructure and if that account was to get compromised, the attackers would have access to all the vital infrastructure. Instead, we have **six separate AWS accounts each hosting a different section of the cloud infrastructure.**

We also use an AWS networking service called AWS Cloud WAN. AWS Cloud WAN makes it easy to build and operate wide area networks that connect your data centers and branch offices, as well as your Amazon Virtual Private Clouds (VPCs).

In this way, we allow the infrastructure to quickly recover in the event of region failure.

Let's look through the architecture steps:

1. A generation site at the edge where distributed energy resources (DER) are located. The DER assets are secured behind a Next Generation Firewall (NGFW). In addition to the Primary and Secondary VPN tunnels from the previous AZ DR scenario that terminate in region 1 (North Virginia), there are two more Primary and Secondary VPN tunnels that terminate in region 2 (Ohio). So, in total there are **four** tunnels, with the following priority:

Priority	Tunnel	Region	AZ	Target	When to use
1	Primary Region 1	us-east-1	us-east-1a	NGFW 1	Region 1 available
2	Secondary Region 1	us-east-1	us-east-1b	NGFW 2	Region 1 available
3	Primary Region 2	us-east-2	us-east-2a	NGFW 3	Region 1 not available
4	Secondary Region 2	us-east-2	us-east-2b	NGFW 4	Region 1 not available

All VPN tunnels are **always active**; however, traffic is sent over only one tunnel, based on the priority. This priority mechanism can be configured in the NGFW console.

The DER site can also deploy Ignition Edge. Ignition Edge is a lightweight version of the Ignition platform developed by Inductive Automation, designed for edge computing applications. It allows users to collect, process, and analyze data from devices and sensors at the edge of a network, closer to where the data is generated. This can help reduce latency, improve response times, and decrease bandwidth usage by minimizing the amount of data sent to a central server or cloud.

A VPN is needed to establish 100% private connectivity from edge to the cloud. If a dedicated **physical connection** from edge to cloud is preferred, AWS Direct Connect can be used.

1.1. The SCADA Control Room, secured behind a Next Generation Firewall (NGFW). The NGFW instance at the SCADA Control Room also makes **four secure IPSEC VPN** tunnels as described in point 1 above.

1.2. The device of the IT Engineer, e.g., a laptop, tablet, etc., on which a client of the Next Generation Firewall (NGFW) is installed. The NGFW client on the device of the IT Engineer can have up to **four secure IPSEC VPN** tunnels as described in point 1 above. Based on the priority, the NGFW client will select the appropriate one.

2. Two instances running the cloud version of the NGFW in a VPC called **Firewall Region-1 VPC**. NGFW-1 is in AZ us-east-1a and NGFW-2 is in AZ us-east-1b. The VPC is part of an AWS account called AWS **Firewall Account 1**.

Using a VPC attachment, the VPC is attached to the AWS Cloud WAN core network, in **region 1, us-east-1.**

3. A Cloud WAN segment called **Security Segment**. A segment is a logical grouping of resources that can be managed together. This helps in organizing and controlling traffic flow between different parts of your network. Segments are dedicated routing domains, which means that, by default, only attachments within the same segment can communicate. A segment is part of a core network, and a core network is part of a global network.

Global network - A single, private network that acts as the high-level container for your network objects. A global network can contain both AWS Transit Gateways and other AWS Cloud WAN core networks.

Core network - The part of your global network managed by AWS. This includes regional connection points and attachments, such as VPNs, VPCs, and Transit Gateway Connects. Your core network operates in the regions that are defined in your core network policy document. In our case, we define the core network across **two regions, i.e., North Virginia (us-east-1) and Ohio (us-east-2).**

The Security Segment is shared with another segment called SCADA Segment. When we create a segment share between two segments, routes from both segments are automatically advertised to each other. So, when we share the Security Segment with the SCADA Segment, VPCs attached to the Security Segment can access resources in VPCs attached to the SCADA Segment. With this setup, the NGFW in **Firewall Region-1 VPC** can securely push traffic to the **Region-1 SCADA VPC.**

4. A segment called SCADA Segment. Using a VPC attachment, this segment is attached to the **Region-1 SCADA VPC**.

This entire AWS Cloud WAN setup is a separate AWS account called the **Connectivity Account**. This account acts as our connectivity platform. All the Firewall and SCADA VPCs in different AWS accounts in region 1 and region 2 connect to the segments, which are part of the core network. This is made possible by a feature called AWS Resource Access Manager or AWS RAM. AWS RAM helps you securely share your resources across AWS accounts. In our case, we use AWS RAM to share the core network with two Firewall accounts in region 1 and 2, two SCADA accounts in region 1 and 2.

5. An internal-facing Application Load Balancer is hosted in two AZs: us-east-1a and us-east-1b. The ALB is part of the **Region-1 SCADA VPC** in Region 1. The Region-2 SCADA VPC is in its separate AWS Account, called **SCADA Account 1**.

6. Ignition Perspective frontend is hosted on an Amazon EC2 instance or on Amazon EKS. Ignition Perspective can access the Ignition Gateway, which is hosted in a different subnet but is part of the same VPC. All subnets in a VPC can natively communicate with each other.

7. The Ignition Backend Gateway is hosted on Amazon EC2. It has a **scheduled backup** configured to store its Gateway Backup file to an Amazon S3 bucket. This frequency to backup can be as low as one minute.

8. The Ignition Database hosted on Amazon Aurora PostgresSQL using the Amazon Aurora Global Database feature. With the Amazon Aurora Global Database, you set up multiple Aurora DB clusters that span multiple AWS regions. Aurora automatically synchronizes all changes made in the primary DB cluster to one or more secondary clusters. An Aurora global database has a primary DB cluster in one region and up to five secondary DB clusters in different regions.

All components in step 5, 6, 7 and 8 are deployed in both AZs, **us-east-1a and us-east-2b, for high availability in region us-east-1, as shown in the AZ DR scenario in the earlier section.**

8.1. In our case, we deploy one Aurora cluster as the Primary in Region 1, us-east-1 and a Secondary cluster in region 2, us-east-2. By using the Amazon Aurora Global Database feature, we ensure that a copy of the database is **always ready and available in region us-east-2 in both AZs us-east-2a and us-east-2b**. In the event of region 1, us-east-1 failure, the Secondary Aurora in region 2, us-east-2, is automatically promoted as the **writer node.**

9. Using VPC Endpoints, a backup of the Primary Ignition configuration file is saved in an Amazon S3 bucket using AWS private backbone.

10. As part of security best practices, the Amazon S3 bucket is in a separate AWS Account called **Backup account.**

We enable the S3 Object lock feature. S3 Object Lock prevents data from being altered or deleted by any person or process, whether unintended or because of malicious activity. This helps prevent ransomware events from deleting or altering your data. We also activate the live CRR or Cross-Region Replication on the S3 bucket. We can use CRR to replicate objects across Amazon S3 buckets in different regions. We use this feature to replicate objects from region us-east-1 to us-east-2 in real time. This ensures there is always a backup of the S3 available in the DR region us-east-2.

To further protect against cyber threats, we can enable S3 encryption. Amazon S3 provides several encryption options to secure your data both at rest and in transit.

Region Disaster Recovery Mechanism

Region disaster recovery is achieved as follows:

2.1. The NGFW present at resources 1, 1.1 and 1.2 identifies that the Primary and Secondary VPN tunnels for region us-east-1 have collapsed due to region failure. At that moment, the Primary and Secondary tunnels take over to send traffic to NGFW 3 and 4 in DR region us-east-2. NGFW 3 in AZ us-east-2a is the Primary Firewall and comes **online first**. If AZ us-east-2a goes offline, then NGFW 4 comes online as part of AZ disaster recovery described earlier.

The NGFW 3 and 4 are in a VPC called **Firewall Region 2 VPC**, in a separate AWS account called **Firewall Account 2.**

3.1. The Security Segment in region us-east-2. Using a VPC attachment, the Firewall Region 2 VPC is attached to the AWS Cloud WAN core network's Security Segment in **region 2, us-east-2.**

4.1 The SCADA Segment in region us-east-2. Using a VPC attachment, this segment is attached to the **Region-2 SCADA VPC**. The Security Segment is shared with the SCADA Segment; so, the NGFW in **Firewall Region-2 VPC** can securely push traffic to the **Region-2 SCADA VPC.**

8.1. By using the Amazon Aurora Global Database feature, we ensure that a copy of the database is **always ready and available in region 2, us-east-2 in both AZs us-east-2a and us-east-2b**. In the event of region 1, us-east-1 failure, the Secondary Aurora in region 2, us-east-2 is automatically promoted as the **writer node.**

One critical component of the Ignition SCADA is the database[TG8] . Using the Amazon Aurora Global Database feature, we have a version of the database in region 2, us-east-2, which is **available and in sync** with the Primary Ignition database, **24x7, 365 days a year.**

11. A region failure is a rare and unlikely event. However, if it occurs, the IT organization will typically declare a major incident, representing region failure. This is usually a priority 1 incident, and the SOP for regional disaster recovery will be initiated.

12. As part of the SOP run book, the disaster recovery agent will run a failover script in AWS System Manager. The failover script will read the latest version of the Primary Ignition configuration file saved in Amazon S3 from step 10. As we had activated CRR or Cross-Region Replication on the S3 bucket, the script will fetch configuration file, from S3 in region 2, us-east-2.

13. Using the configuration file, from Amazon S3 in region 2, us-east-2, the script will launch a CFT or Cloud Formation Template that will deploy the new Ignition Gateway in AZs us-east-2a and us-east-2b to ensure high availability. The Ignition Gateway can then connect with the Ignition Database on Amazon Aurora, which was always available due to the Global Database feature.

14. The script also launches an Amazon EC2 or Amazon EKS to host Ignition Perspective in AZs us-east-2a and us-east-2b.

15. The script also launches an internal-facing Application Load Balancer or ALB to front the Ignition Perspective machines from step 14.

At the end of step 15, all SCADA components, i.e., Frontend Gateways (Perspective), Backend Gateway(s) and Database, [TG9] plus Primary and Secondary VPN tunnels from edge to cloud, have been successfully deployed in **region 2, us-east-2, in both AZs us-east-2a and us-east-2b.**

Using the above setup, a full-scale region disaster recovery can be achieved in a few minutes.

We use the following additional services for auto-scaling, monitoring, security and access management.

- AWS Auto Scaling - Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.
- Elastic Load Balancing - Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones (AZs).
- Amazon GuardDuty - Amazon GuardDuty is a threat detection service that continuously monitors, analyzes, and processes AWS data sources and logs in your AWS environment.
- AWS Identity and Access Management (IAM) - AWS IAM is a web service that helps you securely control access to AWS resources. With IAM, you can manage permissions that control which AWS resources users can access.

- AWS IAM Identity Center - AM Identity Center provides one place where you can create or connect workforce users and centrally manage their access to all of their AWS accounts and applications.
- VPC Flow Logs - VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC.
- Amazon CloudWatch - Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time.

Further reading:

1. Whitepaper - Secure Your Global OT/IT Networks with AWS Cloud WAN
2. Workshop - SCADA Disaster Recovery on AWS for Inductive Automation's Ignition
3. Blog - How SGN extended isolated networks to AWS using AWS Transit Gateway
4. Blog - Simplify global hybrid connectivity with AWS Cloud WAN and AWS Direct Connect integration
5. Blog - Ataccama: Building our global network with AWS Cloud WAN
6. Blog - Build Global SD-WANs with AWS Cloud WAN Tunnel-less Connect
7. AWS Well-Architected Framework

About The Authors



Avneet Singh

Avneet is the EMEA Principal Specialist Industry Architect for OT and Renewables at Amazon Web Services. He is based out of Amsterdam, The Netherlands and is responsible for establishing AWS's leadership position in building resilient cloud native solutions for the Energy vertical. Avneet has more than 15 years of experience in the utility industry, having delivered successful technology solution projects across the meter-to-cash cycle spanning smart metering, billing, invoicing, and regulatory compliance. Avneet has a keen interest in OT/IT convergence, IoT, data analytics, and renewable energy optimization. He is the author of the white paper (solution guidance) on OT/IT Convergence on AWS. He is actively collaborating with energy operators across the world in NAMER, EMEA and APJ regions, developing next-generation OT/IT solutions on the AWS cloud.



Thomas Goetz

Thomas is a Sales Engineer at Inductive Automation, where he helps customers architect, design, and deploy Ignition successfully. With 11 years of manufacturing experience, he originally implemented Ignition for a single process before expanding it across a complex, multi-operation plant. He understands both the technical and budgetary challenges customers face and works to provide solutions that address today's needs while preparing them for future growth and scalability. Beyond his professional role, Thomas has a deep interest in IIoT (MQTT), home automation, and process design. His hands-on experience with Ignition in personal and professional projects gives him unique insight into real-world applications, allowing him to connect with customers on a practical level, helping him to explore innovative solutions and guide customers toward modern, scalable architectures that align with their long-term goals.