Solutions Reference Architecture

April 2024

Accelerate Red Hat® OpenShift® AI Workflows Using Intel's Newest Processor Features with Red Hat® Validated Patterns

Intel solutions architects have extended several Red Hat validated patterns to use Intel® Technology—boosting AI performance[1] and helping protect proprietary data and AI models

Contents

Executive Summary

"AI everywhere" is not just an IT catchphrase. Every enterprise is feeling the impact and benefits that AI workloads are having on their business operations. However, AI can be a bit overwhelming at times. It isn't always easy to know where to start your AI journey or how to reach the next milestone. Red Hat and Intel are working closely together to make it easier to deploy hybrid cloud AI workloads and to accelerate them at runtime.

Red Hat® validated patterns include all the code and Red Hat® OpenShift® elements you need to deploy specific AI use cases. Intel has worked with Red Hat to demonstrate how easy it is to take advantage of specific hardware features built into 5th Generation Intel® Xeon® Scalable processors when using validated patterns. Intel® Advanced Matrix Extensions (Intel® AMX), Intel® Software Guard Extensions (Intel® SGX), and Intel® QuickAssist Technology (Intel® QAT) features are specifically designed to improve AI performance and help fortify security across a wide range of applications and are seamlessly integrated with OpenShift 4.14.

Red Hat validated patterns extended to use Intel® architecture not only accelerate many AI operations but also preserve critical compute resources to help customers cost-effectively reach their AI goals. This reference architecture showcases Intel's enhancement of four Red Hat validated patterns and provides an overview of the necessary tools to design and deploy AI solutions with high performance and advanced security.

Solution Benefits

Extended Red Hat® validated patterns from Intel make it easy to enable AI and security features and accelerations provided by 5th Gen Intel® Xeon® Scalable processors running Red Hat® OpenShift® 4.14:

- Effortless deployment of comprehensive, fully operational AI workflows, eliminating the need for extensive manual configurations.
- Accelerated AI model training and inference with Intel® Advanced Matrix Extensions.
- Enhanced security using Intel® Software Guard Extensions, which supports confidential computing.
- Offloading of encryption and compression to a dedicated accelerator to improve application performance, using Intel® Quick Assistant Technology.

Solution Brief

Business Challenge: Deploying AI Workloads Is Difficult and Time-Consuming

Businesses need to innovate to stay viable. Forward-thinking companies want to deploy new use cases for AI and transition to cloud-native applications running on platforms like Red Hat® OpenShift®. At least 35% of businesses have already adopted some form of AI, and AI will contribute USD 15.7 trillion to the global economy by 2030.[2] However, developers encounter various obstacles when working on AI applications in a hybrid cloud environment. Establishing a stable Kubernetes workflow from the ground up can be difficult due to the number of elements and components involved. Lack of skills and complexity of AI systems are among the top barriers to adopting AI.[3]

Enterprises are asking important questions about how to simplify deploying AI in an OpenShift environment:

- Is there a way to quickly deploy a workload on OpenShift, with minimal configuration and management overhead?
- How can we easily access optimizations that enable fast AI training and inference in the data center or at the edge?
- How can we avoid spending precious CPU cycles on data encryption and decryption?
- Can we encrypt our data at runtime to help keep data and applications secure?

Red Hat and Intel have the answers!

Solution Value: Intel's Extended Validated Patterns Deliver High Performance and Security

Intel has extended several Red Hat® validated patterns to provide unique AI performance and security features. The co-innovation between Red Hat and Intel provides differentiating value through the combination of optimized software plus built-in hardware capabilities.

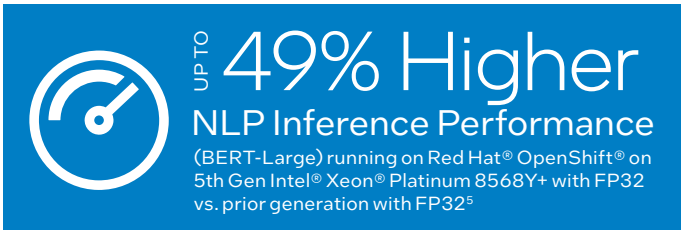
Intel's processor technologies are being integrated with OpenShift using validated patterns to create a seamless experience for Red Hat's customers. 5th Gen Intel® Xeon® Scalable processors use Intel® Advanced Matrix Extensions (Intel® AMX) and Intel® QuickAssist Technology (Intel® QAT) to accelerate AI applications; Intel® Software Guard Extensions (Intel® SGX) helps protect them.

Four New Validated Patterns Enhanced by Intel

Validated patterns are an advanced method for deploying applications in a hybrid cloud. They use a GitOps-based framework to automate the deployment of a full application stack and include all the necessary code and encapsulate deployment best practices.

In early 2024, Intel described two Intel AMX-accelerated validated patterns, based on the Multicloud GitOps and Medical Diagnosis validated patterns.[4] In this reference architecture, we introduce four more patterns that extend the Multicloud GitOps pattern capabilities and showcase the features of 5th Gen Intel Xeon Scalable processors.

- **Multicloud GitOps with OpenShift AI and Intel AMX acceleration.** Building a new AI workflow around your existing applications in a multicluster environment is not easy. Add OpenShift AI to your existing flow so you can start building your own AI system quickly. Also use your favorite AI tools, such as the Intel® Distribution of OpenVINO™ toolkit and accelerate inference by up to 49% by using Intel AMX.[5]
- **Multicloud GitOps with Intel SGX protection.** You worked hard to create your model, and the training data is company property. Protect your application and data using this enhanced pattern, which demonstrates the basics of running containerized applications inside Intel SGX enclaves. Based on this pattern, other applications can be secured with Gramine Shielded Containers (GSC). If you are new to Intel SGX and shielded containers, this is a great pattern to start with.
- **Multicloud GitOps and secrets with Intel SGX protection.** You want to protect tokens, passwords, certificates, API keys, and other critical secrets. This pattern extends the previous one by protecting the HashiCorp Vault, which now runs in an Intel SGX enclave, so that other components of the pattern can use Vault to safely store secrets.
- **Multicloud GitOps accelerated by Intel QAT.** Cryptographic operations are important, but they can use up substantial CPU cycles, negatively affecting AI application performance. This pattern shows how to use the Istio service mesh to offload cryptographic tasks to a hardware accelerator, which can save CPU cycles for AI training and inference. This pattern can be a foundation for deploying more complex distributed applications on the cluster.



Advantages of Validated Patterns

Validated patterns can help developers quickly create a fully functional environment on OpenShift and manage AI workflows reliably and easily, without worrying about configuration details. Developers can use validated patterns—such as those extended by Intel—that include Red Hat® OpenShift® AI. This solution enables developers and data scientists to design more reliable AI workflows faster than ever.[6] OpenShift AI supports the entire AI lifecycle; it enables data acquisition and preparation; model training and fine-tuning; and model serving and model monitoring.

In addition to including all the code a developer needs, validated patterns are continually tested for updates, security, and reliability to promote continuous integration. You can frequently update your application stack with automated builds and tests. This dynamic approach helps improve software quality and reduces validation and release time.

Solution Architecture Highlights

Figure 1 illustrates the solution stack for Intel's extension of Red Hat validated patterns. Through node feature discovery, the patterns can determine if a particular feature, such as Intel AMX or Intel SGX, is available. OpenShift AI, Kubernetes management software, and other pattern components all integrate with OpenShift.

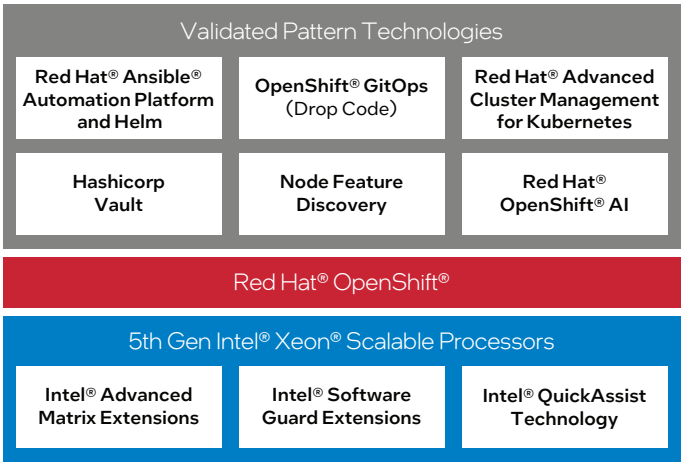


Figure 1. Extending Red Hat® validated patterns to include Intel® technologies can accelerate AI workloads and enhance security.

Key Technologies

The following sections briefly describe some of the most relevant technologies from Red Hat and Intel that power the four extended validated patterns illustrated in this reference architecture.

Red Hat OpenShift

Red Hat OpenShift offers a comprehensive platform powered by Kubernetes to efficiently build, update, and scale applications. You can enhance productivity and speed up the application deployment process by using a full suite of services, all adaptable to your preferred infrastructure.

Red Hat OpenShift AI

Red Hat OpenShift AI serves as a versatile, scalable MLOps platform equipped with tools for constructing, deploying, and managing applications powered by AI. OpenShift AI was developed using open-source technologies: it offers reliable, operationally consistent features for teams to experiment, serve models, and launch innovative applications. OpenShift AI supports the complete lifecycle of AI/ML experiments and models, both on-premises and in the public cloud.

Node Feature Discovery Operator

The Node Feature Discovery (NFD) Operator controls the detection of hardware characteristics and configuration in an OpenShift cluster and labels the nodes with hardware-specific information. It indicates the presence of a technology, accelerator, or module on the host. By determining which nodes have a required hardware feature, DevOps teams can ensure that workflows requiring a specific feature (such as Intel AMX) land on an appropriate node.

Intel Device Plugins Operator

The Intel Device Plugins Operator for OpenShift is a collection of device plugins that advertise specific Intel® hardware resources to the kubelet. For example, the SGX plugin manages the memory allocation pool and ensures that the Intel SGX device nodes are correctly set up within the OpenShift cluster.

5th Generation Intel Xeon Scalable Processors

Designed for AI, 5th Gen Intel Xeon Scalable processors have AI acceleration in every core and are ready to handle your demanding AI workloads, including deep-learning inference and fine tuning on models up to 20 billion parameters.[7] With larger cache, more cores, and support for faster memory, 5th Gen Intel Xeon processors deliver better AI performance than our previous generation[8] unmatched by any other CPU.

Intel Advanced Matrix Extensions

Intel AMX acts as an integrated accelerator that enables 5th Gen Intel Xeon Scalable processors to optimize deep-learning training and inferencing workloads. These processors can swiftly transition between optimizing general computing and AI workloads. Developers have the freedom to code AI functionality using the Intel AMX instruction set, while also coding non-AI functionality using the processor instruction set architecture.

Intel Software Guard Extensions

Intel SGX is a security feature in the Intel Xeon Scalable processor family. It creates an encrypted memory enclave—a trusted environment for processing sensitive data and application code, protecting against malicious software (even if it has admin privileges) and unauthorized access from other applications and processes running on the same host. Intel SGX allows developers to use CPU instructions to increase access control; prevent data modification and disclosure; and enhance code security. Intel SGX is particularly useful in contexts where confidential computing is essential, providing increased security for sensitive and mission-critical data and application code.

Intel QuickAssist Technology

Intel QAT is a built-in workload accelerator on Intel Xeon Scalable processors. It enhances performance and conserves crucial computing resources by offloading key tasks such as data compression and decompression; encryption and decryption; and public key data encryption from the CPU cores. This technology not only accelerates these operations but also contributes to improved overall system performance.

Intel® oneAPI Deep Neural Network Library

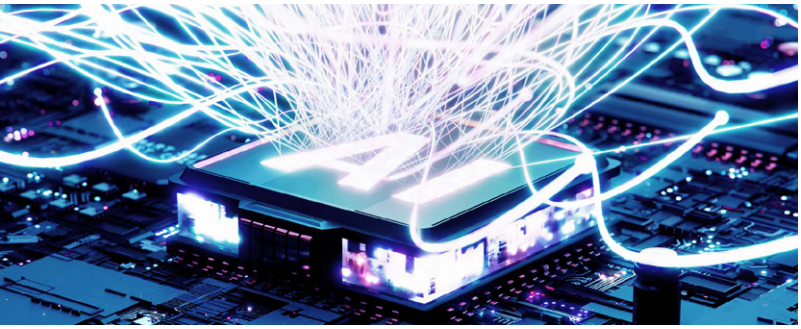
The Intel oneAPI Deep Neural Network Library (oneDNN) provides highly optimized implementations of deep-learning building blocks. It is an open-source, cross-platform library that abstracts instruction sets and other complexities of performance optimization away from the actual code. Developers can use oneDNN to improve the performance of frameworks already in use, such as OpenVINO toolkit, AI Tools from Intel, PyTorch, and TensorFlow. They can also more quickly develop deep-learning applications and frameworks using optimized building blocks.

Gramine Shielded Containers (GSC)

GSC allows a lift-and-shift approach for converting existing container images to a protected version. Essentially, the existing application runs in an encrypted enclave. You can use Intel SGX with GSC with no code changes.

Istio

Istio is an open-source service mesh that provides a uniform and efficient way to secure, connect, and monitor services in a distributed or microservices architecture. It works by adding a proxy "sidecar" along with every deployed application, allowing for application-aware traffic management, observability, and robust security capabilities. Istio's features include secure service-to-service communication with TLS encryption; automatic load balancing; fine-grained control of traffic behavior; and automatic metrics, logs, and traces for all traffic within a cluster. Istio is designed for extensibility and can handle a diverse range of deployment needs. It runs on Kubernetes but can also extend the mesh to other clusters or connect VMs.



Use Cases

The following sections provide an overview of how to use the four validated patterns that are enhanced by Intel® Technology. They also provide links to the patterns themselves and to additional documentation.

Multicloud GitOps with OpenShift AI and Intel AMX Acceleration

This pattern provides a scalable and flexible solution that can adapt to your evolving AI needs. It shows how to use the power of OpenShift AI to enhance your existing workflows, thereby speeding up designing, building, and integrating your AI system in a multicluster environment.

Moreover, whether you're looking to optimize your current applications or develop new AI-powered solutions, this pattern offers a streamlined approach to achieve your objectives and accelerates your workflow with Intel AMX.

The basic Multicloud GitOps pattern has been extended with OpenShift AI and the Intel Distribution of OpenVINO toolkit to highlight the capabilities of the 5th Gen Intel Xeon Scalable processors. It offers developers a streamlined pathway to accelerate their workloads through the integration of AMX, providing efficiency and performance optimization for AI workloads.

- **OpenShift AI:** OpenShift AI serves as a robust platform for creating AI-driven applications and provides a collaborative environment for data scientists and developers that can help them move easily from experiment to production. It is integrated with Jupyter Notebooks, with pre-configured environments and necessary support and optimizations (such as CUDA, PyTorch, TensorFlow, Habana AI, etc.). The validated pattern provides developers with OpenShift AI that's fully configured and ready to go.
- **OpenVINO:** The OpenVINO Toolkit Operator manages OpenVINO components within the OpenShift environment. The OpenVINO Model Server (OVMS) is a scalable, high-performance solution for serving machine-learning models optimized for Intel® architectures. Another component in this pattern is a Jupyter Notebook resource. This element integrates Jupyter from OpenShift AI with a container image that includes developer tools from the OpenVINO toolkit. It also enables users to select a defined image OpenVINO toolkit from the Jupyter Spawner choice list.

Example: There's a BERT-Large example included in the pattern. It is a widely-known model used by various enterprise Natural Language Processing (NLP) workloads. Intel has demonstrated that **5th Gen Intel Xeon Scalable processors deliver up to 49% better NLP performance versus the prior-generation processor.**[9]

Pattern Prerequisites

To run this pattern, you will need:

- An OpenShift 4.14 cluster with dynamic StorageClass to provision Persistent Volumes. The pattern was tested with OpenShift Data Foundation.
- A cluster with workers equipped with 5th Gen Intel Xeon Scalable processors.

Implementation Details

For 5th Gen Intel Xeon Scalable processors, the kernel detects Intel AMX at runtime, so there is no need to enable and configure it additionally to improve performance. However, tools and frameworks that are optimized to take advantage of Intel AMX acceleration are required, such as the OpenVINO toolkit.

All the components required for successfully running AI workloads with Multicloud GitOps pattern are automatically installed when you deploy the pattern. The source code for the extended pattern, Multicloud GitOps with OpenShift AI, and Intel AMX Acceleration is in the repository.

To start the deployment process on your OpenShift cluster, you should fork the repository, create your own branch, and make any required changes. All of the details about installing the extended pattern are in the documentation.

Once the pattern is installed, make sure that all the operators are ready to use. Follow the pattern's instructions for setting up the OpenVINO toolkit notebook correctly. This notebook is your working space where you can develop your own solution. Once the operators and notebook are set up, use the next section to run the BERT-Large example to test if everything is working properly.

Running the BERT-Large Example

1. Set up a oneDNN environment variable in the verbose state, so you can verify in the logs that Intel AMX is present, accessible, and used in the workload. In this example, the variable is named `ONEDNN_VERBOSE`.

```
%env ONEDNN_VERBOSE=1
```

2. Download the BERT-Large model (compatible with FP32 and BF16 precision) named `bert-large-uncased-whole-word-masking-squad-0001`.

```
!omz_downloader –name bert-large-uncased-whole-word-masking-squad-0001
```

3. Go to the directory with the downloaded model and run the benchmark tool with the flag `infer_precision bf16` to use BF16 precision.

```
%cd /opt/app-root/src/intel/bert-large-uncased-whole-word-masking-squad-0001/FP32/
!benchmark_app -m bert-large-uncased-whole-word-masking-squad-0001.xml -infer_precision bf16
```

4. In the `ONEDNN_VERBOSE` variable, you should see an `avx_512_core_amx` entry, which confirms that Intel AMX instructions are being used. The entry is highlighted in the following screen capture as part of the log.

```
[ INFO ] Benchmarking in inference only mode (inputs filling are not included in measur
ement loop).
onednn_verbose,exec,cpu,reorder,jit:uni,undef,src_f32::blocked:abc::f0 dst_bf16::blocke
d:acb::f0,,,1x384x1024,0.471924
onednn_verbose,exec,cpu,reorder,jit:uni,undef,src_f32::blocked:abc::f0 dst_bf16::blocke
d:acb::f0,,,1x384x1024,0.197021
onednn_verbose,exec,cpu,reorder,jit:uni,undef,src_bf16::blocked:acb::f0 dst_bf16::block
ed:abc::f0,,,1x384x1024,0.105957
onednn_verbose,exec,cpu,inner_product,brgemm:avx512_core_amx,forward_inference,src_bf1
6::blocked:ab::f0 wei_bf16:a:blocked:AB16b64a2b::f0 bia_f32::blocked:a::f0 dst_bf16::bl
ocked:ab::f0,attr-scratchpad:user ,,mb384ic1024oc1024,0.758057
onednn_verbose,exec,cpu,inner_product,brgemm:avx512_core_amx,forward_inference,src_bf1
6::blocked:ab::f0 wei_bf16:a:blocked:AB16b64a2b::f0 bia_f32::blocked:a::f0 dst_bf16::bl
ocked:ab::f0,attr-scratchpad:user attr-post-ops:eltwise_linear:0.125 ,,mb384ic1024oc102
4,0.593994
onednn_verbose,exec,cpu,inner_product,brgemm:avx512_core_amx,forward_inference,src_bf1
6::blocked:ab::f0 wei_bf16:a:blocked:AB16b64a2b::f0 bia_f32::blocked:a::f0 dst_bf16::bl
ocked:ab::f0,attr-scratchpad:user ,,mb384ic1024oc1024,0.597168
onednn_verbose,exec,cpu,matmul,brgemm:avx512_core_amx,undef,src_bf16::blocked:abcd::f0
wei_bf16::blocked:abcd::f0 dst_bf16::blocked:abcd::f0,attr-scratchpad:user ,,1x16x384x6
4:1x16x64x384,0.444092
```

5. If you'd like to run an additional example using this same instance of OpenShift AI, follow this example.

Multicloud GitOps with Intel SGX Protection

Intel SGX is a security technology that helps protect your data from disclosure or modification at runtime. Intel implemented two extended validated patterns—Hello World and Vault with Intel SGX—to showcase that Intel SGX plays a crucial role in protecting operations, especially in a high-risk application. The rest of this section focuses on the Hello World example; the Vault with Intel SGX pattern is described later.

The Hello World example provides a basic introduction to the world of Intel SGX. This example shows how to add an extra layer of protection to a GitOps workflow in the simplest possible way. It is a good starting point if you want to create your own validated pattern or extend an existing one with Intel SGX.

Implementation Details

The basic Multicloud GitOps pattern has been extended by adding three components (helm charts): Intel Device Plugins Operator, NFD Operator, and a simple Python application (*`hello-world-sgx`*). The `hello-world-sgx` application uses an image converted with GSC tools to include Intel SGX-related information and execute the application inside an enclave using the Gramine Library OS. This is an easy way to convert an application to a secure version. The sample image is available in the Red Hat container registry and is pulled automatically during the pattern installation process with default variables. When an application is run in an Intel SGX protected enclave using GSC, it adds a header to the logs of running applications.

Because the `hello-world-sgx` application must be running on the node with a CPU supporting Intel SGX, the NFD and Intel Device Plugins Operator are deployed as a part of this extended pattern. NFD is a requirement for Intel Device Plugins Operator installation. Both operators are installed automatically during pattern deployment through subscriptions provided in the `values-hub.yaml` file. Templates for those charts have been copied from Intel Technology Enabling for OpenShift repository.

NFD configuration contains creating two custom resources:

- NodeFeatureDiscovery named `nfd-instance`. This allows the use of labels identifying Intel SGX and exposes the node's SGX Enclave Page Cache (EPC) memory section size.
- NodeFeatureRule named `intel-dp-devices`. This includes rules for assigning those labels based on hardware features. Confirmation of successful configuration is a new label on the node with Intel SGX enabled in BIOS: `intel.feature.node.kubernetes.io/sgx=true`.

Intel Device Plugins Operator configuration involves creating the custom resource, `SgxDevicePlugin`, which allows pods to claim and consume the Intel SGX EPC memory.

The job template for the `hello-world-sgx` pod contains the following section with Intel SGX resources to be scheduled on the proper node:

```
resources:
  requests:
    sgx.intel.com/epc: "1Gi"
  limits:
    sgx.intel.com/epc: "1Gi"
```

Pattern Prerequisites

To run this pattern, you will need:

- An OpenShift 4.14 cluster with dynamic StorageClass to provision Persistent Volumes. The pattern was tested with ODF and LVM Storage solutions.
- A cluster with at least one worker machine that has Intel SGX enabled in BIOS to run protected workloads. Depending on your platform, the enablement of Intel SGX might vary. Refer to your motherboard or server documentation for Intel SGX enablement. Make sure that at least 2 GB of RAM is allocated to Intel SGX (`PRMRR Size` settings). In the sample settings, a higher value was allocated but this can be adjusted to your platform.
- As a prerequisite to prepare an image protected with GSC, you must have a machine with the same OS as the input image. In this case, it is Ubuntu 22.04. This machine does not need to have Intel SGX enabled; it is used for conversion purposes only. Other prerequisites for GSC are described in Gramine's documentation.
- Also install this validated pattern's dependencies.

Installing the Pattern

There are two options to install this extended pattern:

- Using the OpenShift web console
- Using the command line

Both methods are described in the pattern installation documentation; choose the way that suits you best. The pattern deployment procedure is described on the pattern website.

After pattern deployment, when the `hello-world-sgx` pod that's running in the `hello-world-sgx` namespace changes status to Completed, the logs of the `hello-world-sgx` pod will display the "HelloWorld!" message, preceded by information about starting Gramine. The message, as shown here, confirms that Intel SGX has been used to secure the application.

```
2 lines
1  Gramine is starting. Parsing TOML manifest file, this may take some time...
2  HelloWorld!
```

Multicloud GitOps and Secrets with Intel SGX Protection

HashiCorp Vault is an excellent example of an application which needs to be run in a protected enclave, because it stores and processes secrets that are used by other validated pattern components. Intel SGX supports both user-level and OS-level code to define private regions of memory whose contents are protected and unable to be either read or saved by any process outside the enclave itself, including processes running at higher privilege levels.

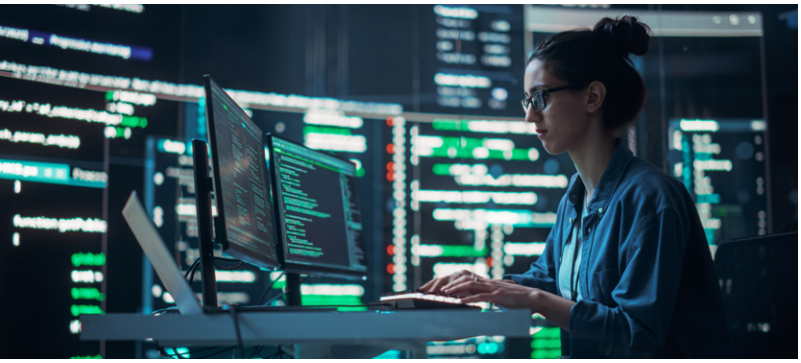
By using Intel SGX, a Vault-based validated pattern can provide a higher level of protection for your cluster's secrets. From now on, they are protected not only in transport, but also in Vault's memory at runtime. Vault was chosen as a component to be protected, because it is a common component of the Multicloud GitOps pattern, and it may be used in many other patterns. Therefore, learning how to use Intel SGX to secure Vault in this pattern can have a multiplier effect across many of your validated pattern use cases.

Implementation Details, Requirements, and Pattern Installation

This extended pattern builds on the basic Intel SGX validated pattern described earlier. Both the NFD and Intel Device Plugins Operator are deployed as a part of the pattern.

For the enhanced Vault pattern, there's an extra requirement in addition to the Hello World pattern's prerequisites. To run this pattern, you will need to use a special protected Vault image that is available from the Red Hat container registry. If you would like to build such an image on your own, there are step-by-step instructions available in the pattern's documentation.

Otherwise, all implementation and pattern installation details described for the Hello World example also apply to this Intel SGX with Vault pattern.



Multicloud GitOps Accelerated by Intel QAT

This extended pattern provides a scalable and extendable cryptography solution for your OpenShift cluster. The pattern's workflow deploys every necessary component needed for Intel QAT to work properly. It also deploys a simple demo based on the Istio service mesh to showcase the usage of Intel QAT's acceleration capabilities. The entire solution can be extended by adding new services that communicate with each other.

Implementation Details

The basic Multicloud GitOps pattern has been extended by four components: three operators—Intel Device Plugins, NFD, and Sail—and a simple HTTP request-and-response application called `httpbin`. The primary purpose of this extension is to enable TLS handshaking and showcase how Intel QAT accelerates cryptographic operations.

The Sail Operator deploys the Istio service mesh. It is basically an extra layer added on top of user applications that adds capabilities like traffic management between services and security, without manually adding them to the code. The version of Istio deployed by the Sail Operator allows for additional configuration, that is, enabling Intel QAT for cryptographic operations. Istio proxy sidecar containers use the `PrivateKeyProvider` framework, which handles TLS handshake private key operations, signing, and decryption. This extended validated pattern modifies the configuration of the Istio proxy to offload these TLS operations to Intel QAT, which creates the possibility for CPU performance benefits.

The `httpbin` application is an HTTP request-and-response service used for testing and showcase purposes. The workload is a standard one used in Istio examples of how to set up a secure TLS ingress gateway.

Installing the Pattern

Before deploying this extended validated pattern, configure the BIOS on all worker nodes to enable Intel QAT according to official instructions. Please consult your platform guide for details. Configuration options should be similar to these:

- Socket configuration > IIO configuration > Intel® VT-d Enabled
- Platform Configuration > Miscellaneous Configuration > SR-IOV Support Enabled
- Socket Configuration > IIO Configuration > IOAT Configuration > Sck<n> > IOAT Configuration Enabled

Once the BIOS is configured properly, you can start the validated patten deployment process, according to the documentation.

This validated pattern uses the Machine Config Operator to apply all the necessary kernel command-line and OS configuration changes. For the example workload, Machine Config objects are used to enable IOMMU and VFIO passthrough and set MEMLOCK to unlimited for all the worker nodes in the cluster. The user does not need to manually set these options.

After the pattern finishes configuring the worker nodes, the next step is to deploy Istio. In the pattern, the service mesh is specifically configured to offload cryptographic operations in TLS handshakes to Intel QAT, freeing CPU cores to perform other work. The modified Istio configuration looks like this:

```
meshConfig:
 defaultConfig:
  privateKeyProvider:
   qat:
    pollDelay: 5ms
sidecarInjectorWebhook:
 templates:
  custom: |
   spec:
    containers:
    - name: istio-proxy
     securityContext:
      capabilities:
       add:
        [IPC_LOCK]
      privileged: false
      allowPrivilegeEscalation: false
     resources:
      limits:
       qat.intel.com/cy: 1
      requests:
       qat.intel.com/cy: 1
```

The highlighted blocks of the YAML code show the most important changes:

- Sets the `PrivateKeyProvider` field to the QAT device, which delegates TLS handshake cryptographic operations to Intel QAT.
- Adds the `IPC_LOCK` capability, which is required by the QAT device, to securityContext. The validated pattern also creates a Security Context Constraint called `restricted-v2-istio`, which is mandatory for `IPC_LOCK`.
- Assigns QAT cryptographic resources (`qat.intel.com/cy`) to every Istio proxy sidecar container.

You can verify if the configuration is properly loaded by looking at the logs of every pod with an Istio proxy sidecar container. The appearance of QAT as the private key provider in the Istio ingress gateway pod logs confirms that the Intel QAT configuration is loaded (assuming there are no errors appearing).

```
5012","privateKeyProvider":{"qat":{"pollDelay":"0.005s"}}}
```

```
31    configPath: ./etc/istio/proxy
32    controlPlaneAuthPolicy: MUTUAL_TLS
33    discoveryAddress: istiod-istio-system.istio-system.svc:15012
34    drainDuration: 45s
35    privateKeyProvider:
36    qat:
37    pollDelay: 0.005s
38    proxyAdminPort: 15000
39    serviceCluster: istio-proxy
40    statNameLength: 189
```

After Istio is properly configured, the last step of the pattern installation is deploying the `httpbin` application deployment. This application returns a simple 418 code response if the client sends the correct HTTPS. If a user attempts to access the `httpbin` service with the `curl` tool using the wrong certificate chain (specified through the `cacert` flag), the connection will not be established and will simply fail. All generated keys and certificates are stored in two places:

- The directory `/home/<USERNAME>/certs`
- The Vault deployed on the OpenShift cluster

To send a HTTPS request, run the following command:

```
export INGRESS_HOST=<WORKER_NODE_IP>
export INGRESS_NS=istio-system
export INGRESS_NAME=istio-ingressgateway
export SECURE_INGRESS_PORT=$(oc -n "${INGRESS_NS}"
get service "${INGRESS_NAME}" -o jsonpath='{.spec.
ports[?(@.name=="https")].nodePort}')

curl --resolve "httpbin.example.com:${SECURE_
INGRESS_PORT}:$INGRESS_HOST" --cacert certs/
httpbin.example.com.crt "https://httpbin.example.
com:${SECURE_INGRESS_PORT}/status/418"
```

If the client receives a 418 code as a response, that means that Istio and the `httpbin` service are working as intended.

Summary

Businesses that are successfully deploying AI to automate manual tasks, improve customer experience, manage cyber risk, and uncover new actionable insights are undoubtedly gaining a competitive edge over companies that are slow to innovate. However, complex interrelationships between infrastructure components and an AI skills gap make gaining AI traction difficult. 93% of US and UK organizations consider AI to be a business priority and have projects planned or already in production. However, more than half of them (51%) acknowledge that they don't have the right mix of skilled AI talent in-house to bring their strategies to life.[10]

Red Hat and Intel are collaborating to simplify and accelerate the deployment of hybrid cloud AI workloads, with a focus on data security and intellectual property protection. Red Hat has developed validated patterns—essentially, proven infrastructure recipes—for deploying specific use cases, while Intel is continuing to demonstrate the simplicity of extending validated patterns to use Intel technologies for improved performance and enhanced security.

The combination of Red Hat's easy-to-use solutions, like OpenShift AI and various operators, and hardware features in 5th Gen Intel Xeon Scalable processors make using these validated patterns nearly a push-button experience. Key hardware features include:

- Intel AMX for better training and inference performance.
- Intel SGX for better security.
- Intel QAT for offloading CPU-intensive cryptography operations.

These features are integrated with OpenShift 4.14 and are showcased through the extended Red Hat validated patterns described in this reference architecture. Intel and Red Hat will continue to co-innovate to integrate the hardware and software components you need for AI use cases, while simplifying the deployment and management of the solution stack.

Learn More

- 5th Generation Intel® Xeon® Scalable processors
- Intel® Advanced Matrix Extensions
- Intel® Software Guard Extensions
- Intel® QuickAssist Technology
- Red Hat validated patterns
- Boosting Business with AI article
- Level Up Your NLP applications on Red Hat OpenShift and 5th Gen Intel® Xeon® Scalable Processors blog

Contact your Intel or Red Hat representative today to discuss how the synergy between Red Hat validated patterns and Intel Technology can benefit your business.

Revision History

Revision Number	Description	Date
v1.0	Version 1.0	April 2024

[1] See https://community.intel.com/t5/Blogs/Tech-Innovation/Data-Center/Level-Up-Your-NLP-applications-on-Red-Hat-OpenShift-and-5th-Gen/post/1572320
[2] Authority Hacker, January 2024, "149 AI Statistics: The Present and Future of AI [2024 Stats]."
[3] CompTIA, February 2024, "Top Artificial Intelligence Statistics and Facts for 2024."
[4] See https://validatedpatterns.io/patterns/multicloud-gitops-amx/ and https://validatedpatterns.io/patterns/medical-diagnosis-amx/, respectively.
[5] See endnote 1.
[6] See https://www.redhat.com/en/technologies/cloud-computing/openshift/openshift-ai.
[7] Based on Intel internal modeling as of December 2023.
[8] See [A15-A16] at intel.com/processorclaims: 5th Gen Intel Xeon Scalable processors. Results may vary.
[9] See endnote 1.
[10] SnapLogic, "The AI Skills Gap."