

Step 1: Clone the Repository

```
``` bash
git clone https://github.com/ndeepakprasanth/dapr-eks-pubsub-ready.git
cd dapr-eks-pubsub-ready
```

## **Step 2: Set execute permissions**

```
chmod +x oneclick.sh
chmod +x test.sh
chmod +x scripts/*.sh
```

## **Step 3: One-Click deployment**

```
Replace with your AWS Account ID
ACCOUNT_ID=946248011760 ./oneclick.sh
```

## **Step 4: Verify deployment**

```
Check pods (should show 2/2 Ready)
kubectl -n dapr-apps get pods -o wide
Check services
kubectl -n dapr-apps get svc
Check Dapr components
kubectl -n dapr-apps get components
```

## **Step 5: Test Pub/Sub functionality**

```
./test.sh
```

## **Step 6: View ECR repositories**

```
aws ecr describe-repositories --region us-east-1
```

## **Step 7: Clean up when completed**

```
./destroy.sh
```

### 1. Running pods:

```
423946@AMBGB000622 dapr-eks-pubsub-ready % kubectl -n dapr-apps get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
orderservice-874558c9b-q88hz 2/2 Running 0 4m41s 192.168.47.173 ip-192-168-36-44.ec2.internal <none> <none>
productservice-787cc7d9c7-8ztmz 2/2 Running 0 4m41s 192.168.26.113 ip-192-168-13-87.ec2.internal <none> <none>
423946@AMBGB000622 dapr-eks-pubsub-ready %
```

### 2. Services:

```
423946@AMBGB000622 dapr-eks-pubsub-ready % kubectl -n dapr-apps get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
orderservice ClusterIP 10.100.48.232 <none> 8090/TCP 48m
orderservice-dapr ClusterIP None <none> 80/TCP,50001/TCP,50002/TCP,9090/TCP 48m
productservice ClusterIP 10.100.230.157 <none> 8080/TCP 48m
productservice-dapr ClusterIP None <none> 80/TCP,50001/TCP,50002/TCP,9090/TCP 48m
```

### 3. Dapr components

```
423946@AMBGB000622 dapr-eks-pubsub-ready % kubectl -n dapr-apps get components
NAME AGE
snsssqs-pubsub 6m15s
```

#### 4. Test Pub/Sub

```
423946@AMBGB000622 dapr-eks-pubsub-ready % ./test.sh | tee FinalOutput.rtf
==> Testing Dapr pub/sub functionality
Publishing test message...
% Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 132 100 76 100 56 1291 951 ---:---:---:---:---:---:---:--- 2275
{"ok":true,"published":{"orderId":999,"item":"test-laptop","price":1299.99}}pod "test-curl" deleted
==> Verifying delivery in OrderService logs...
Defaulted container "orderservice" out of: orderservice, daprd
Delivery OK: OrderService received the event.

==> OrderService logs:
Defaulted container "orderservice" out of: orderservice, daprd
OrderService listening on 8090
OrderService received event: {"data":{"item":"test-laptop","orderId":999,"price":1299.99},"datacontenttype":"application/json","id":"a220ce1f-f09a-423a-bd44-ec1f9f8e9392","pubsubname":"snssqs-pubsub","source":"productservice","specversion":"1.0","time":"2025-12-23T14:30:25Z","topic":"orders","traceid":"","traceparent":"","tracestate":"","type":"com.dapr.event.sent"}
OrderService received event: {"data":{"item":"test-laptop","orderId":999,"price":1299.99},"datacontenttype":"application/json","id":"9140340d-80ee-4d5f-83d5-c90d80a328ad","pubsubname":"snssqs-pubsub","source":"productservice","specversion":"1.0","time":"2025-12-23T14:46:07Z","topic":"orders","traceid":"","traceparent":"","tracestate":"","type":"com.dapr.event.sent"}

==> Pod status:
NAME NOMINATED NODE READY STATUS RESTARTS AGE IP NODE
orderservice-76c489798c-9gntr ec2.internal <none> 2/2 Running 0 16m 192.168.22.177 ip-192-168-3-230.
productservice-5bd6fb54ff-m4pkk ec2.internal <none> 2/2 Running 0 23m 192.168.41.34 ip-192-168-43-22.

==> Services:
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S)
AGE
orderservice ClusterIP 10.100.142.39 <none> 8090/TCP
27m
orderservice-dapr ClusterIP None <none> 80/TCP,50001/TCP,50002/TCP,9090/TCP
27m
productservice ClusterIP 10.100.156.239 <none> 8080/TCP
27m
productservice-dapr ClusterIP None <none> 80/TCP,50001/TCP,50002/TCP,9090/TCP
27m

==> Dapr components:
NAME AGE
snssqs-pubsub 28m
423946@AMBGB000622 dapr-eks-pubsub-ready %
```

## 5. ECR repositories with pushed images

```
423946@AMBGB000622 dapr-eks-pubsub-ready % aws ecr describe-repositories --region us-east-1 --profile Deepak
{
 "repositories": [
 {
 "repositoryArn": "arn:aws:ecr:us-east-1:946248011760:repository/orderservice",
 "registryId": "946248011760",
 "repositoryName": "orderservice",
 "repositoryUri": "946248011760.dkr.ecr.us-east-1.amazonaws.com/orderservice",
 "createdAt": "2025-12-22T11:41:09.817000+00:00",
 "imageTagMutability": "MUTABLE",
 "imageScanningConfiguration": {
 "scanOnPush": false
 },
 "encryptionConfiguration": {
 "encryptionType": "AES256"
 }
 },
 {
 "repositoryArn": "arn:aws:ecr:us-east-1:946248011760:repository/productservice",
 "registryId": "946248011760",
 "repositoryName": "productservice",
 "repositoryUri": "946248011760.dkr.ecr.us-east-1.amazonaws.com/productservice",
 "createdAt": "2025-12-22T11:41:06.238000+00:00",
 "imageTagMutability": "MUTABLE",
 "imageScanningConfiguration": {
 "scanOnPush": false
 },
 "encryptionConfiguration": {
 "encryptionType": "AES256"
 }
 }
]
}
423946@AMBGB000622 dapr-eks-pubsub-ready %
```

## 6. Bedrock analysis

```
423946@AMBGB000622 dapr-eks-pubsub-ready % python3 bedrock.py
== Bedrock (Titan) Analysis ==
[{'tokenCount': 116, 'outputText': 'snsqs-pubsub 62m\npod "test-curl" deleted\n\nThe logs indicate that the pub/sub test was successful. The OrderService is listening on port 8090, and the Pod status shows that both the OrderService and ProductService are running. The services are exposed externally with ClusterIPs and have the appropriate ports open. The Dapr components, including the SNSSQS pub/sub component, are also running.\n\nThere are no errors or warnings in the logs. The pub/sub test was successful.', 'completionReason': 'FINISH'}]
423946@AMBGB000622 dapr-eks-pubsub-ready %
```

## 7. Stress test output

```
423946@AMBGB000622 dapr-eks-pubsub-ready % echo "==== STRESS TEST DEMONSTRATION ==="
echo "Sending multiple concurrent requests..."
for i in {1..5}; do
 kubectl -n dapr-apps run load-test-$i --restart=Never --image=curlimages/curl:8.10.1 --command -- curl -X POST http://productservice:8080/publish -H 'Content-Type: application/json' -d "{\"orderId\": $i, \"item\":\"load-test-$i\", \"price\": $(RANDOM % 1000)}" &
done
echo "Waiting for completion..."
sleep 8
echo "==== STRESS TEST RESULTS ==="
kubectl -n dapr-apps get pods | grep load-test
echo "==== CLEANUP ==="
kubectl -n dapr-apps delete pods -l run --field-selector=status.phase=Succeeded
==== STRESS TEST DEMONSTRATION ===
Sending multiple concurrent requests...
[2] 7386
[3] 7387
[4] 7388
[5] 7389
[6] 7390
Waiting for completion...
pod/load-test-5 created
pod/load-test-4 created
[6] + done kubectl -n dapr-apps run load-test-$i --restart=Never --command -- curl -X
[5] + done kubectl -n dapr-apps run load-test-$i --restart=Never --command -- curl -X
pod/load-test-1 created
[2] + done kubectl -n dapr-apps run load-test-$i --restart=Never --command -- curl -X
pod/load-test-3 created
[4] + done kubectl -n dapr-apps run load-test-$i --restart=Never --command -- curl -X
pod/load-test-2 created
[3] + done kubectl -n dapr-apps run load-test-$i --restart=Never --command -- curl -X

==== STRESS TEST RESULTS ===
load-test-1 0/1 Completed 0 8s
load-test-2 0/1 Completed 0 8s
load-test-3 0/1 Completed 0 8s
load-test-4 0/1 Completed 0 8s
load-test-5 0/1 Completed 0 8s

==== CLEANUP ===
pod "load-test-1" deleted
pod "load-test-2" deleted
pod "load-test-3" deleted
pod "load-test-4" deleted
pod "load-test-5" deleted
```

## 8. Logs screenshot:

```
423946@AMBG800622: dapr-eks-pubsub-ready % echo "==" ARCHITECTURE VERIFICATION =="
echo ""
echo "■ EKS Cluster Nodes:"
kubectl get nodes -o wide
echo ""
echo "■ Dapr System Pods:"
kubectl -n dapr-system get pods
echo ""
echo "■ Application Logs (ProductService):"
kubectl -n dapr-apps logs deploy/productservice --tail=5
echo ""
echo "■ Application Logs (OrderService):"
kubectl -n dapr-apps logs deploy/orderservice --tail=5
== ARCHITECTURE VERIFICATION ==

■ EKS Cluster Nodes:
NAME STATUS ROLES AGE VERSION
ip-192-168-13-87.ec2.internal Ready <none> 164m v1.32.9-eks-ecaa3a6 192.168.13.87 100.49.41.36 Amazon Linux 2023.9.20251208 6.1.158-180.294.amzn2023.x86_
64 containerd://2.1.5
ip-192-168-36-44.ec2.internal Ready <none> 164m v1.32.9-eks-ecaa3a6 192.168.36.44 54.196.115.105 Amazon Linux 2023.9.20251208 6.1.158-180.294.amzn2023.x86_
64 containerd://2.1.5

■ Dapr System Pods:
NAME READY STATUS RESTARTS AGE
dapr-operator-69845db889-4bpm6 1/1 Running 0 162m
dapr-placement-server-0 1/1 Running 0 162m
dapr-scheduler-server-0 1/1 Running 0 162m
dapr-scheduler-server-1 1/1 Running 0 162m
dapr-scheduler-server-2 1/1 Running 0 162m
dapr-sentry-78f477df9c-wg7p 1/1 Running 0 162m
dapr-sidecar-injector-6f774f9bfc-c54cw 1/1 Running 0 162m

■ Application Logs (ProductService):
Defaulted container "productservice" out of: productservice, daprd
 metadata: [Object],
 reason: 'DAPR_PUBSUB_PUBLISH_MESSAGE'
}

■ Application Logs (OrderService):
Defaulted container "orderservice" out of: orderservice, daprd
OrderService listening on 8090
```