



IBM MQ v9.x Best Practices for Economies of Scale

Presentation by

John Carr, Sr. Consultant at TxMQ

Presentation Overview

- **Presentation Based on Experience I had Migrating from MQ v7.x to v9.0.0.x LTS on distributed in several places**
 - ▶ We'll cover the best practices we employed along the way
- **Presentation Section 1**
 - ▶ Compatibility (Cipher specs, JREs, MQ Clients, etc.)
 - ▶ Cross-Section of MQ v9 in Production
 - ▶ Walk Through a Simple Migration Activity
- **Presentation Section 2**
 - ▶ Organizing
 - Team
 - Environment
 - ▶ Server Build Scenario

Section 1

1.1 Compatibility

MQ Client Compatibility

- **“WebSphere MQ V7.0, V7.1, V7.5, V8.0, V9.0 and V9.1**
queue managers and clients inter operate with queue managers and clients from any previous level of the WebSphere MQ or MQSeries products”
- **In other words**
 - ▶ Version 7.x, 8.0 and 9.x clients can connect to version 7.x, 8.0 and 9.x queue managers
 - ▶ Version 7.x, 8.0 and 9.x queue manager can interact with version 7.x, 8.0 and 9.x clients
 - ▶ Version 7.x, 8.0 and 9.x queue manager can interact with version 7.x, 8.0 and 9.x queue managers

<http://www-01.ibm.com/support/docview.wss?uid=swg21312967>

SSL Cipher Spec and Cipher Suites for TLS 1.2

■ WebSphere Application Server 7.x and 8.5

- ▶ Where MQ Resource Adaptor Level at 7.0 or 7.5
 - **CipherSpec on MQ CHANNEL** - TLS_RSA_WITH_AES_256_CBC_SHA256
 - **CipherSuite on WAS** - SSL_RSA_WITH_AES_256_CBC_SHA256
- ▶ Where MQ Resource Adaptor Level at 8.0
 - **CipherSpec on MQ CHANNEL** - ECDHE_RSA_AES_256_CBC_SHA384
 - **CipherSuite on WAS** - SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384

NOTE: App teams using WAS 7.x/8.5 were encouraged to upgrade to WAS 9

■ WebSphere Application Server 9

- ▶ Where MQ Resource Adaptor Level at 8.0 or 9.0
 - **CipherSpec on MQ CHANNEL** - ECDHE_RSA_AES_256_CBC_SHA384
 - **CipherSuite on WAS** - SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384

NOTE: MQ 9.0 resource adapter bundled with WAS 9 (fix pack ??)

SSL Cipher Spec and Cipher Suites for TLS 1.2

■ Standalone Java

▶ Where MQ Java Client Library at 7.0 or 7.5

- **CipherSpec on MQ CHANNEL** - TLS_RSA_WITH_AES_256_CBC_SHA256
- **CipherSuite on Oracle JRE** - TLS_RSA_WITH_AES_256_CBC_SHA256

▶ Where MQ Java Client Library at 8.0 or 9.0

- **CipherSpec on MQ CHANNEL** - ECDHE_RSA_AES_256_CBC_SHA384
- **CipherSuite on Oracle JRE** -
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

■ Other Details

- ▶ **Ensure the IBM or Oracle JRE is at v7 or v8 with appropriate fix packs**
- ▶ **In surprise, this was the biggest “gotcha” we encountered**

<https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct>

1.2 Other Considerations

CD vs LTS

■ Long Term Support (LTS)

- ▶ Demands the highest levels of stability
- ▶ Releases approximately every two years
- ▶ Release is supported for at least five years

■ Continuous Delivery (CD)

- ▶ Intended for customers wanting to explore the latest features
- ▶ Released every few months
- ▶ When moving to a newer CD version, un-install the binaries for the previous CD
 - QMGR Objects and Defs should not be impacted
- ▶ Release is supported for the lifetime of the related LTS

CD vs LTS

- Decision of using CD or LTS, depending on the size of your supported MQ Network, should be taken with caution
- It will drive how you build and maintain your environments (Dev, Test, etc.)
- In a nutshell, decision was made to stick with LTS because we needed stability (and predictability)

CD vs LTS



- Though the decision was to use LTS for our MQ network, we did setup three virtual servers and put into a “parking lot” to reserve to explore CD features, if necessary.

In-Place Install vs. Side by Side vs. “Fresh Install”

■ In-Place Install

- ▶ Installing v9 as an upgrade to the v7 installation on the same server (replacing the binaries)
- ▶ Afterward, when starting the QMGR, it should be migrated to the new version

■ Side by Side Install

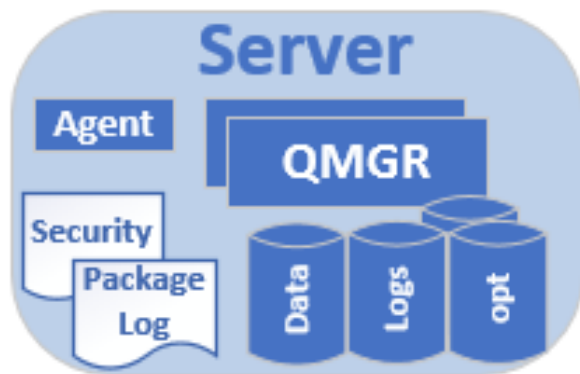
- ▶ Have two code bases on the same server (I.E. v7 and v9)
- ▶ Use the setmqm command to associate the QMGR code base

■ “Fresh Install”

- ▶ New, provisioned server with MQ v9 codebase installed
- ▶ Will have to create a new QMGR and “copy” object defs from existing QMGR to new

In-Place Install vs. Side by Side vs. “Fresh Install”

■ The decision? “Fresh Install”



- ▶ This was chosen most often because it's a good idea to get server refreshes every few years.
- ▶ Requires extra work and more help from other operations teams, but it's always good because it gave the opportunity to “clean house” with the QMGRs

Naming Standards

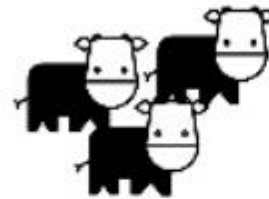
Cattle, not pets!



Has a personalized name

Unique, Special and Cared for

Taken care when ill



Have numbers assigned

Are usually identical

Replaced when not in good health

You need more, you buy more

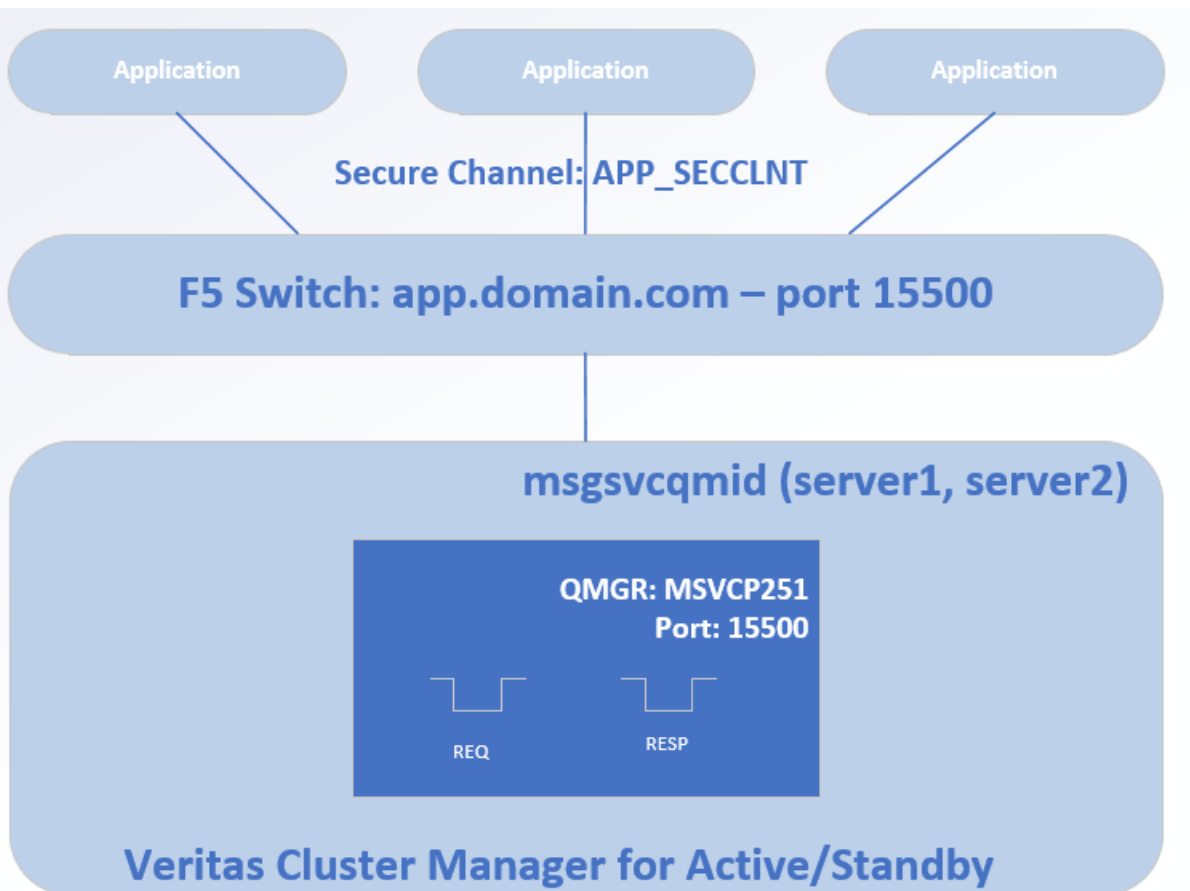
Src: Gavin McCance from CERN

Flux7 

Regardless of your MQ network size, it's best to treat them like cattle, not pets. This includes QMGR and Object names.

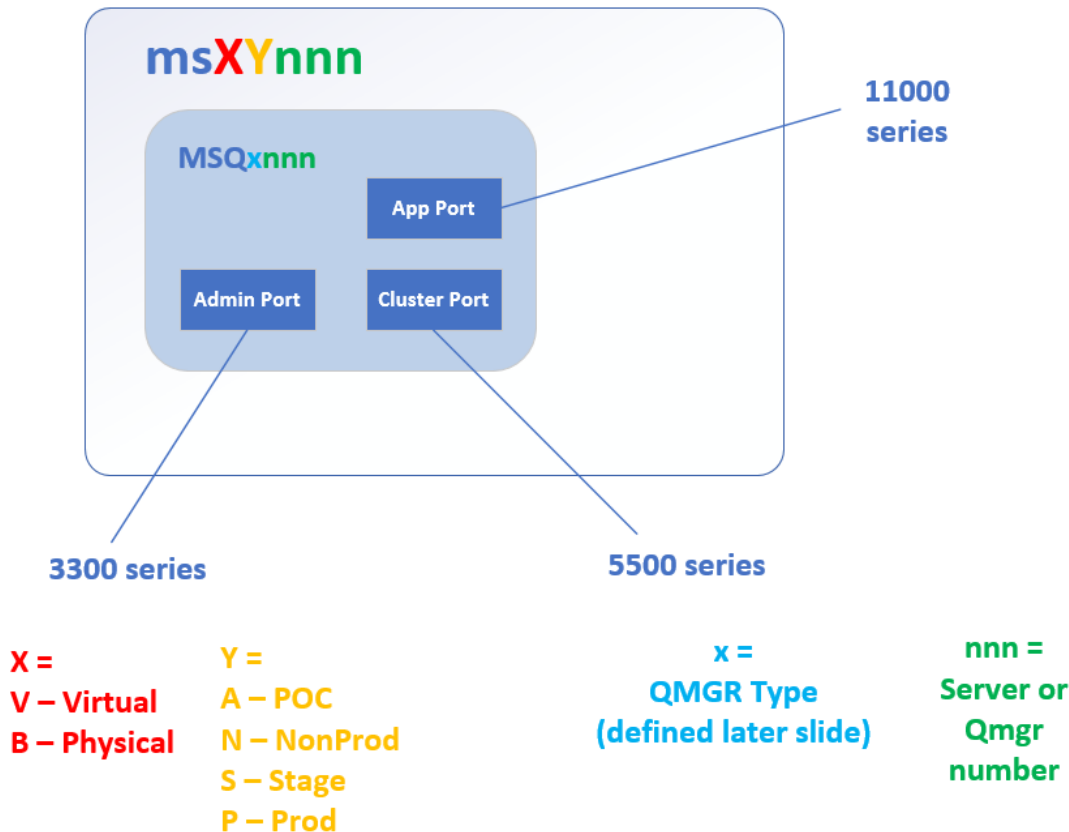
1.3 Cross-Section of MQ v9 in Production

Messaging Service in a Nutshell



- Applications use a secure channel to connect to QMGR
- LDAP Authorization/Authentication (cover more soon)
- Apps connect to F5
- Veritas used as HA (active/standby)
- Effort was to hide details about true QMGR location

QMGR Listener Ports



■ Followed a strategy where three ports would be employed

- ▶ Application Port
 - Application traffic
 - Shared environment still used a single listener port for app traffic
- ▶ Cluster Port
 - MQ Cluster Traffic
- ▶ Admin Port
 - Internal use (if needed)

QMGR Organization

QM Numbering Scheme

MSQ^{xnnn}

100 series – DMZ

500 series – CORE

900 series – POC

- Follow a strategy for numbering your QMGRs.
- It helps to follow a range meant for different operational domains: **CORE, DMZ, POC**
- **Especially for LDAP**
 - ▶ Most shops have at least two LDAP servers
 - DMZ
 - CORE
 - ▶ Different LDAP operational domains will drive a different setup for AUTHINFO and AUTHRECS

QMGR Listener Ports

```
DEFINE LISTENER ('MSQP504_15103') TRPTYPE(TCP) CONTROL(QMGR) +  
    PORT(15103)  +  
    IPADDR('msvc504')  +  
    DESCR('App Listener Port for ... ')  +  
    ...
```

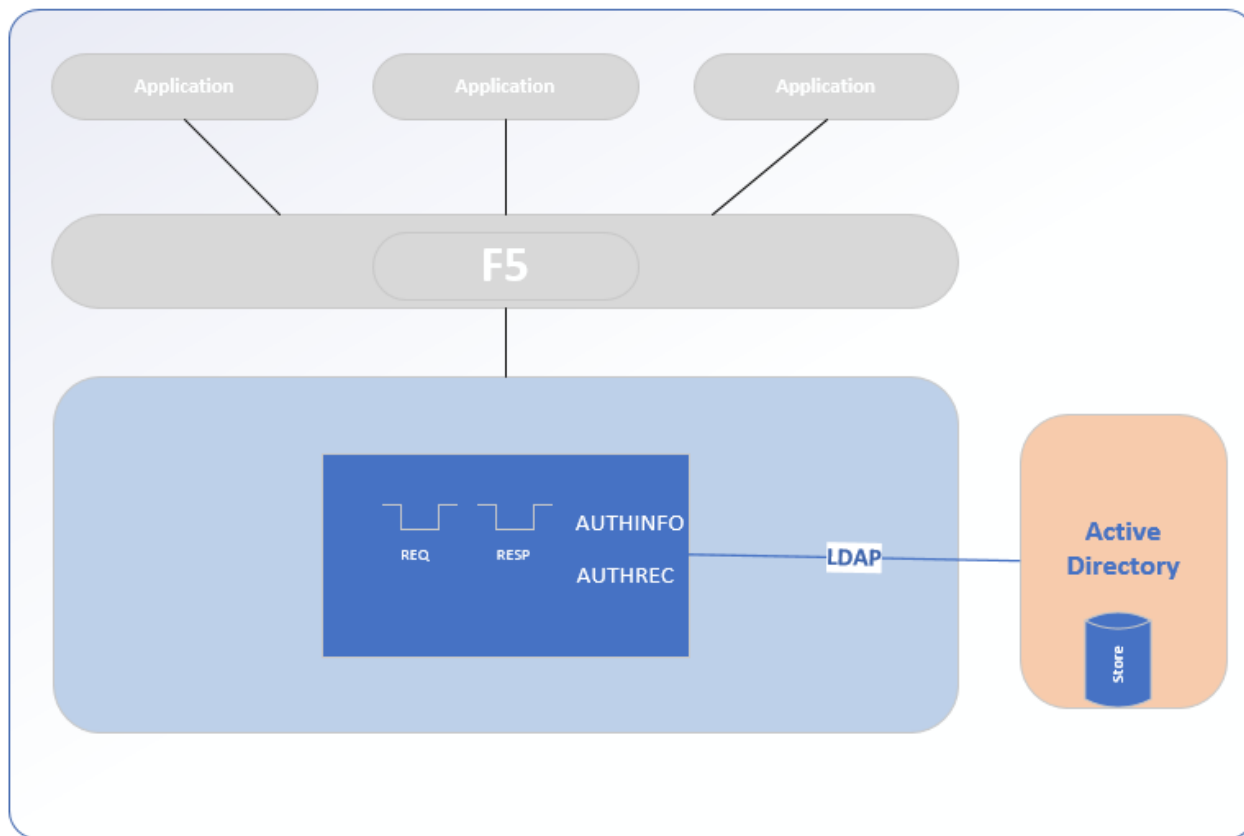
- **Cattle, not Pets!**
 - ▶ Come up with a strategy where your MQ Objects for LISTENERs, CHANNELs, or where it makes sense to give a “branding name” that uniquely identifies the object
- **<QMGR>_<PORT>**
- **Don't forget a useful DESCR – Really!!**

CHANNELS

```
DEFINE CHANNEL('INVENTORY_SECCLNT')  +  
    CHLTYPE(SVRCONN)  +  
    DESCR('Inventory App SSL Client')  +  
    ...  
    SSLCAUTH(REQUIRED)  +  
    SSLCIPH(ECDHE_RSA_AES_256_CBC_SHA384')  +  
    TRPTYPE(TCP)  +  
    ...
```

- Each App has it's own CHANNEL to connect
- Don't let Apps share CHANNELs, especially when dealing with SSLCIPHs
- Don't forget a DESCR
- <APPNAME>_SECCLNT

Authorization and Authentication



■ Authentication

- ▶ AUTHINFO
- ▶ Definitions used to connect to an LDAP Server

■ Authorization

- ▶ AUTHREC
- ▶ Definitions for what user/group can have access to which MQ Objects

■ Most shops will have at least two LDAP/AD servers

- ▶ CORE
- ▶ DMZ

Active Directory (AD)

- **Define a Windows Global group access specifically for use with MQ**
 - ▶ Ideally, create two groups: NON-PROD and PROD
- **Define a windows application id and associate it with the group**
 - ▶ Ideally, create two IDs: NON-PROD and PROD
- **Provide the id/group information to MQ admin for authorization**
 - ▶ MQ admins could use ldapsearch on the userID from command line and inspect the results
- **NOTE: When accessing a MQ v9 queue manager the defined windows application id does not need login rights to the hosting MQ server**

Group ID and user ID suggested format:

Group = xxxmsvtgrp (test) xxxmsvpgrp (prod)

ID = xxxmsvtst (test) xxxmsvprd (prod)

Where **xxx**: application acronym that will be used as the queue high level qualifier

Example of ID/group names as:

Non-Prod environments: user id: **abcmsvtst**, group id: **abcmsvtgrp**

Production Environment: user id: **abcmsvprd**, group id: **abcmsvpgrp**

NOTE: **12 character restriction** for group user ids used with MQ

AUTHINFO

```
DEFINE AUTHINFO(ATH.PW1) +  
  AUTHTYPE(IDPWLDAP) +  
  CONNAME('ad-ldap-prod-...') +  
  SHORTUSR('...') +  
  ADOPTCTX(YES) +  
  AUTHORMD(SEARCHUSR) +  
  BASEDNG('CN=Users,DC=ms,DC=ds,...') +  
  BASEDNU('CN=Users,DC=...') +  
  LDAPUSER('CN=...') +  
  CLASSGRP('group') +  
  LDAPPWD('secret password - shh!')  
  ...
```

```
ALTER QMGR CONNAUTH(ATH.PW1)  
REFRESH SECURITY(*)
```

```
DEFINE AUTHINFO(ATH.PW2) +  
  AUTHTYPE(IDPWLDAP) +  
  CONNAME('ad-ldap-prod-...') +  
  SHORTUSR('...') +  
  ADOPTCTX(YES) +  
  AUTHORMD(SEARCHUSR) +  
  BASEDNG('CN=Users,DC=ms,DC=ds,...') +  
  BASEDNU('CN=Users,DC=...') +  
  LDAPUSER('CN=...') +  
  CLASSGRP('group') +  
  LDAPPWD('secret password - shh!') +  
  ...
```

- Create two AUTHINFO objects
- Assign one to QMGR CONNAUTH
- Passwords change often – 60 to 90 days
- Good practice to do this because you can schedule better password changes with a second AUTHINFO
- Less service interruption

AUTHREC

SET AUTHREC

```
>>-SET AUTHREC--+-+-----+-----+-----+
      '-PROFILE--(--profile-name--)-'
```

```
>---OBJTYPE--(--+AUTHINFO--+)-+-----+
      +-CHANNEL--+
      +-CLNTCONN--+
      +-COMMINFO--+
      +-LISTENER--+
      +-NAMELIST--+
      +-PROCESS--+
      +-QUEUE-----+
      +-QMGR-----+
      +-RQMNAME--+
      +-SERVICE--+
      '-TOPIC-----'
```

```
>---PRINCIPAL--(--principal-name--)-+-----+
      '-GROUP--(--group-name--)------'
```

AUTHADD-- (-----NONE-----+---)

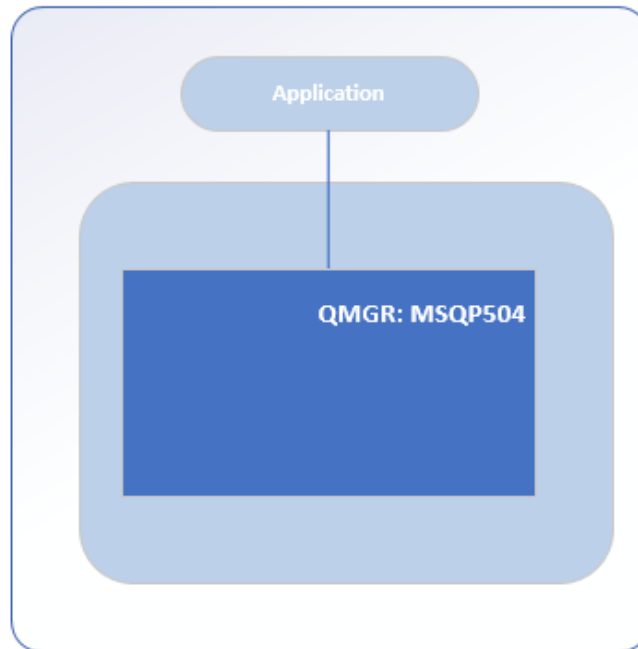
```
+-ALTUSR--+
+-BROWSE--+
+-CHG-----+
+-CLR-----+
+-CONNECT--+
+-CRT-----+
+-DLT-----+
+-DSP-----+
+-GET-----+
+-INQ-----+
+-PUT-----+
+-PASSALL--+
+-PASSID--+
+-SET-----+
+-SETALL--+
+-SETID--+
+-SUB-----+
+-RESUME--+
+-PUB-----+
+-SYSTEM--+
+-CTRL-----+
+-CTRLX---+
+-ALL-----+
+-ALLADM--+
'-ALLMQI--'
```

- **Fine Grain vs Coarse Gran Access**
- **Fine Grain**
 - ▶ Specific access – only what's needed by the app
 - ▶ GET or PUT only for UserID
- **Coarse Grain**
 - ▶ Cast a wide net for access to a group of objects
 - ▶ All queues beginning with APP1.** get ALLMQI for UserGroup
- **Combinations thereof...**

AUTHREC

- **Use a common sense approach to security**
 - ▶ Give an application authorization for only what it needs
 - ▶ What is the application's purpose?
- **For Example: a business app needs**
 - ▶ Connect to a QMGR
 - ▶ GET messages from queues starting with APP.IN.*
 - ▶ BROWSE messages from another app's local queue called APP2.AUDIT.QUEUE
 - ▶ PUT messages to queue called APP.APP2.PO.QUEUE
 - ▶ PUBLISH to a TOPIC Object APP_TOPIC
 - ▶ Be a SUBscriber or RESUME a subscription to another TOPIC Object called APP_TOPIC2
- **The more you know about an app using a QMGR, the better you can grant only the necessary access**

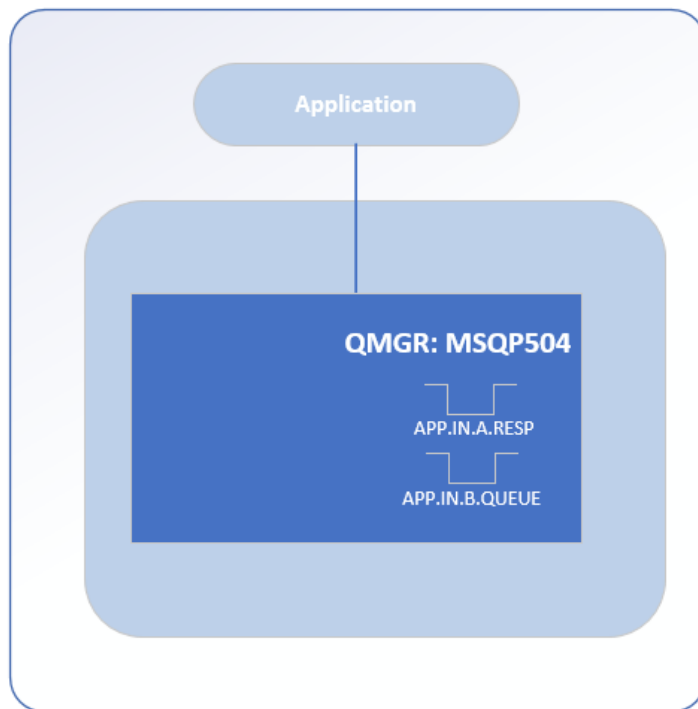
AUTHREC: Connect to QMGR



- A business app's needs
 - ▶ Connect to a QMGR
- Use a PROFILE and give the object name is the QMGR name
- Use the AD User Group when granting access

```
SET AUTHREC PROFILE(MSQP504) OBJTYPE(QMGR) GROUP('CN=appgrp') AUTHADD(INQ,CONNECT)
```

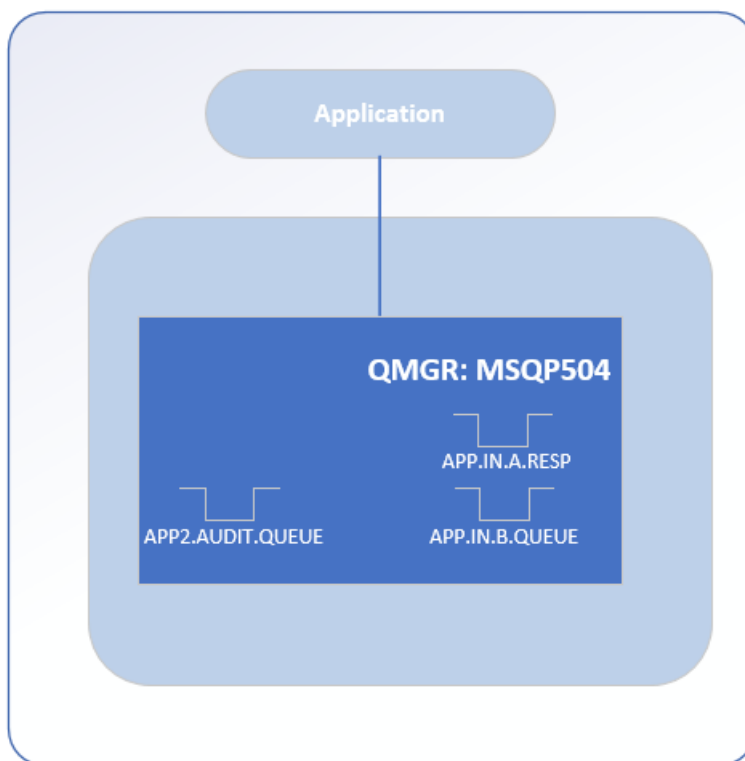
AUTHREC: GET Message from Queue



- **A business app's needs**
 - ▶ GET messages from queues starting with APP.IN.*
- **Use a PROFILE and give APP.IN.** as it's name**
- **Meant for LOCAL queues**
- **BEST PRACTICE**
 - ▶ Use wildcards (*) only for queues meant as input to a specific app

```
SET AUTHREC PROFILE(APP.IN.** ) OBJTYPE(Queue) GROUP('CN=appgrp') AUTHADD(INQ,GET)
```

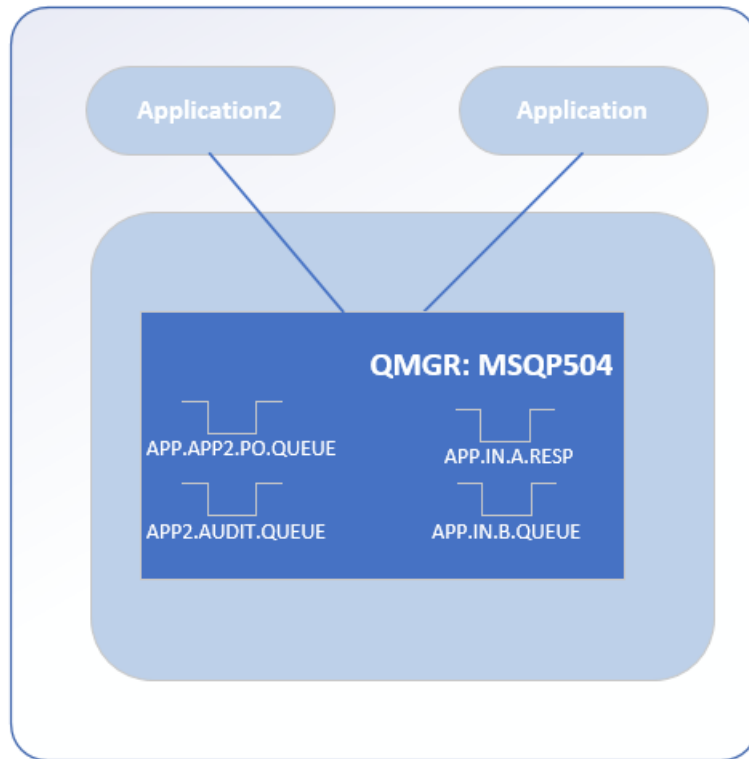
AUTHREC: BROWSE a Queue



- **A business app's needs**
 - ▶ BROWSE messages from another app's local queue called APP2.AUDIT.QUEUE
- **Use a PROFILE and give the actual queue object as it's name**
- **Meant for LOCAL queues**
- **BEST PRACTICE**
 - ▶ Be specific when granting and NOT use wild cards (*) when browsing another app's queue

```
SET AUTHREC PROFILE(APP2.AUDIT.QUEUE) OBJTYPE(Queue) GROUP('CN=appgrp') AUTHADD(INQ,BROWSE)
```

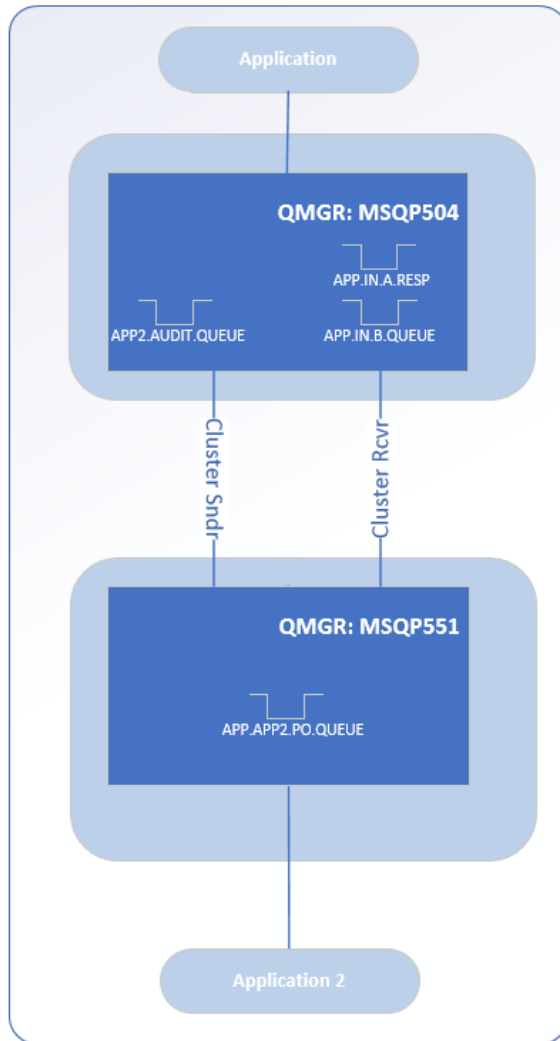
AUTHREC: PUT to a Queue



- **A business app's needs**
 - ▶ PUT messages to queue called APP.APP2.PO.QUEUE
- **Use a PROFILE and give the actual queue object as it's name**
- **Meant for LOCAL, REMOTE and clustered queues**
- **BEST PRACTICE**
 - ▶ Be specific when granting and NOT use wild cards (*) because...

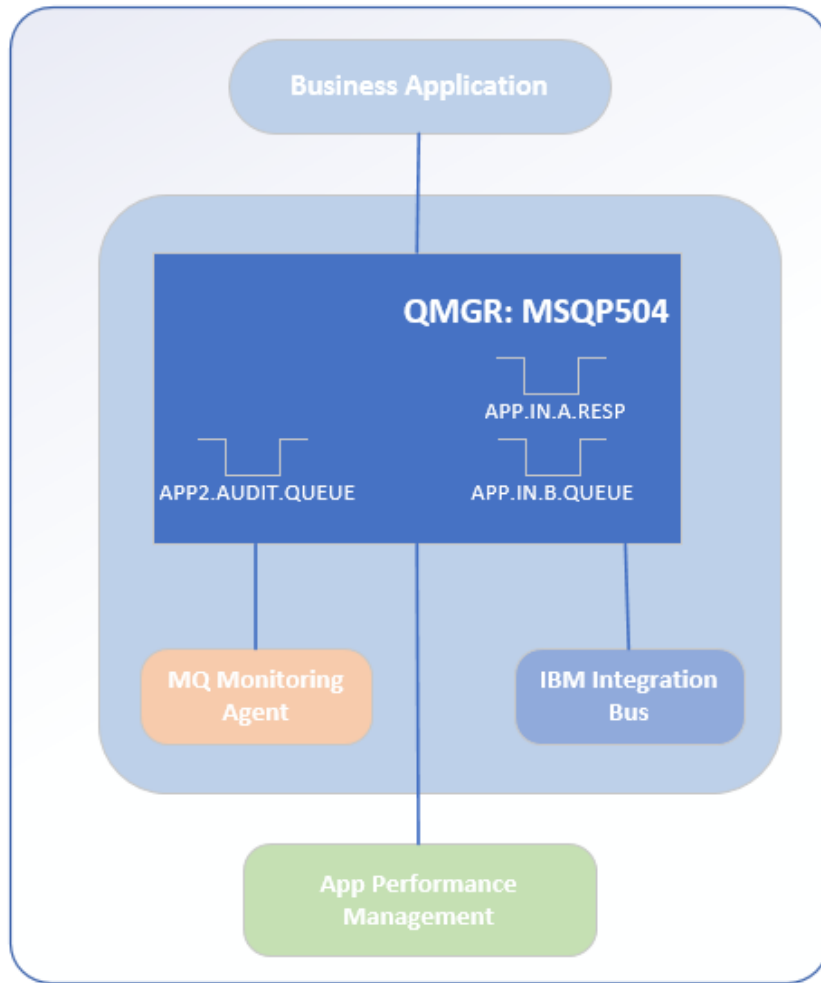
```
SET AUTHREC PROFILE(APP.APP2.PURCHASE.ORDER.QUEUE) OBJTYPE(Queue) GROUP('CN=appgrp') AUTHADD(PUT)
```

AUTHREC: PUT to a Queue



- **Because...**
 - ▶ You may need to move app2 to another QMGR someday and use a cluster queue
 - ▶ When Specific with the AUTHREC PUT permissions, then you don't need to change
- **If App1 wants to put a message to a queue, it really shouldn't have to know the location of that queue**

Other Considerations



■ BEST PRACTICE

- ▶ Start thinking about anything connecting to a queue manager as an application
- ▶ Don't let apps run as the MQ owner (mqm)

■ Application Types

- ▶ Business App
- ▶ ACE or IIB
- ▶ MQ Monitoring Agent
- ▶ App Performance Mgt Product

■ They'll need to have AUTHRECs set for them, too

■ Doing this, you'll have a healthy, happy MQ ecosystem 😊

1.4 Simple Migration Activity

Queue Manager Object Inventory for App

<i>MQ 7.5 QMGR</i>	<i>Shared by APP1 and APP2</i>
ALIAS	LOCAL
APP2.APP1_EFUNDS_REQ	APP1.EFUNDS_REQ
APP1.APP2_EFUNDS_RESP	APP2.EFUNDS_RESP
APP1.APP2_PO_REQ	APP2.PO_REQ
APP2.APP1_PO_RESP	APP1.PO_RESP
APP2.APP1_PAYFEEDBACK	APP1.PAYFEEDBACK
APP1.APP2_SHIP.STATUS	APP2.SHIP.STATUS

- APP1 and APP2 are shared on the same QMGR
- APP1 team is ready to migrate to MQ v9, but APP2 team won't be ready till 2019.
- Take an MQ Object inventory of all queues and topics used by APP1 and review with the app team
- It's your opportunity to remove un-used queues or topics

Migrate Queues

<i>MQ 7.5 QMGR</i>	<i>APP2</i>	<i>MQ 9.x QMGR</i>	<i>APP1</i>
ALIAS	LOCAL	ALIAS	LOCAL
		APP2.APP1_EFUNDS_REQ	APP1.EFUNDS_REQ
APP1.APP2_EFUNDS_RESP	APP2.EFUNDS_RESP		
APP1.APP2_PO_REQ	APP2.PO_REQ		
		APP2.APP1_PO_RESP	APP1.PO_RESP
		APP2.APP1_PAYFEEDBACK	APP1.PAYFEEDBACK
APP1.APP2_SHIP.STATUS	APP2.SHIP.STATUS		

Set AUTHREC to PUT (APP2)
APP2.APP1_EFUNDS_REQ
APP2.APP1_PO_RESP
APP2.APP1_PAYFEEDBACK

Set AUTHREC to PUT (APP1)
APP1.APP2_EFUNDS_RESP
APP1.APP2_PO_REQ
APP1.APP1_SHIP.STATUS

Migrate Queues

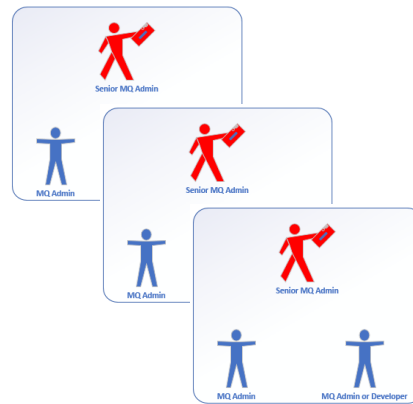
<i>MQ 7.5 QMGR</i>	<i>APP2</i>	<i>MQ 9.x QMGR</i>	<i>APP1</i>
ALIAS	LOCAL	ALIAS	LOCAL
		APP2.APP1_EFUNDS_REQ	APP1.EFUNDS_REQ
APP1.APP2_EFUNDS_RESP	APP2.EFUNDS_RESP		
APP1.APP2_PO_REQ	APP2.PO_REQ		
		APP2.APP1_PO_RESP	APP1.PO_RESP
		APP2.APP1_PAYFEEDBACK	APP1.PAYFEEDBACK
APP1.APP2_SHIP.STATUS	APP2.SHIP.STATUS		

Set AUTHREC to GET (APP2)
APP2.EFUNDS_RESP
APP2.PO_REQ
APP2.SHIP.STATUS

Set AUTHREC to GET (APP1)
APP1.EFUNDS_REQ
APP1.PO_RESP
APP1.PAYFEEDBACK

Section 2

2.1 Organizing Your Team



Organizing Your Team

- In surprise, this is the biggest challenge: Where to start?
- Divide into three groups
 - ▶ Leadership Group
 - ▶ Server Build Group
 - ▶ Application Enablement Group

Organizing Your Team

MQ Leadership Group



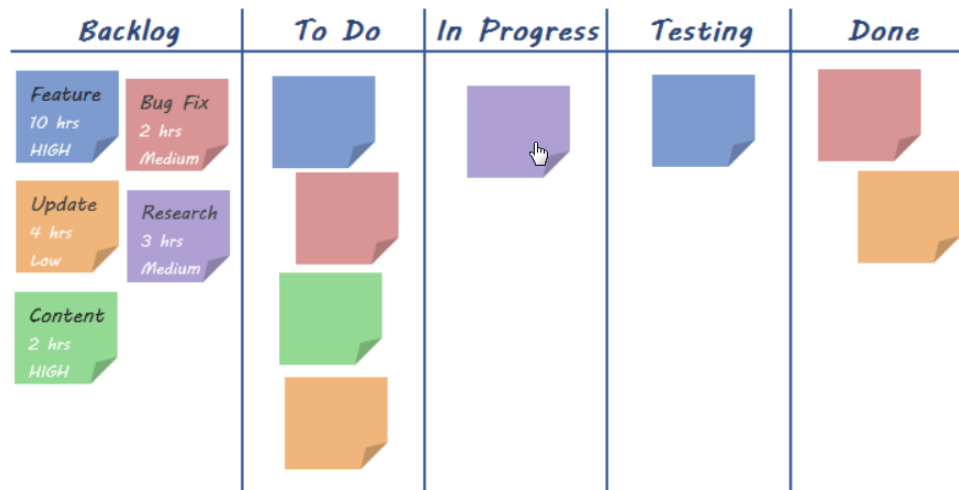
Server Build Group



Application Enablement Group



Organizing Your Team



- Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively. SOURCE: VersionOne – What is Kanban
- Have the entire team participate in a 15 minute “stand-up” every morning to review the tasks and identify bottlenecks early.
- Your CIO will love teams adopting this into their culture!

Example for Organizing Your Teams

Separation of Duties

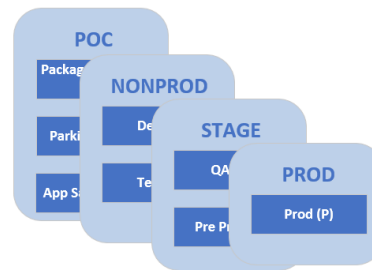
Server Build Activities

- Kernel ParmS
- Ulimit Settings
- Filesystems
- MQ Server Install
- HA Clustering (STAGE/PROD)
- Job Scheduling
- QMGR Configs
- Monitoring Agent Configs
- Certificate management
- etc.

Application Activities

- Connectivity Port for App
- Coordination with F5 (STAGE/PROD)
- Firewall Request Rules
- Connect to MQ Cluster
- Coordination with App Team
- Custom monitoring rules
- Decommissioning old qmgr
- etc.

2.2 Organizing Your Environment



Organizing Your Environment

- Always remember the basics for standards
 - ▶ Server Naming
 - ▶ QMgr Naming
 - QMgr Range Numbering
 - Port Range Numbering
 - ▶ Environment Designations
 - ▶ Virtual or Physical Servers
 - You may not be ready for containers or the cloud
 - ▶ Etc.

Organizing Your Environment



X =

V – Virtual

B – Physical

Y =

A – POC

N – NonProd

S – Stage

P – Prod

x =

QMGR Type
(defined later slide)

nnn =

Server or
Qmgr
number

Organizing Your Environment

QM Numbering Scheme

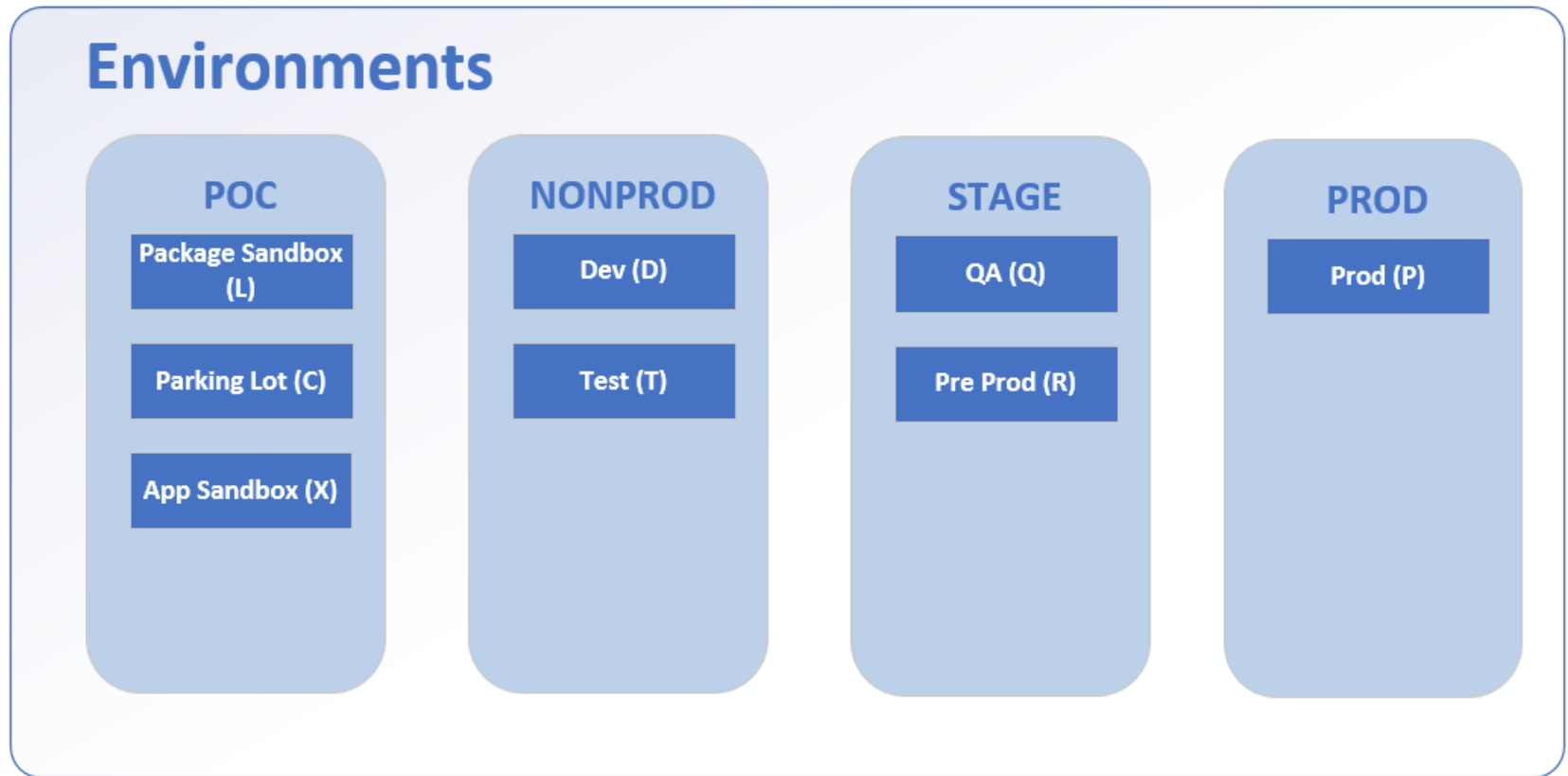
MSQxnnn

100 series – DMZ

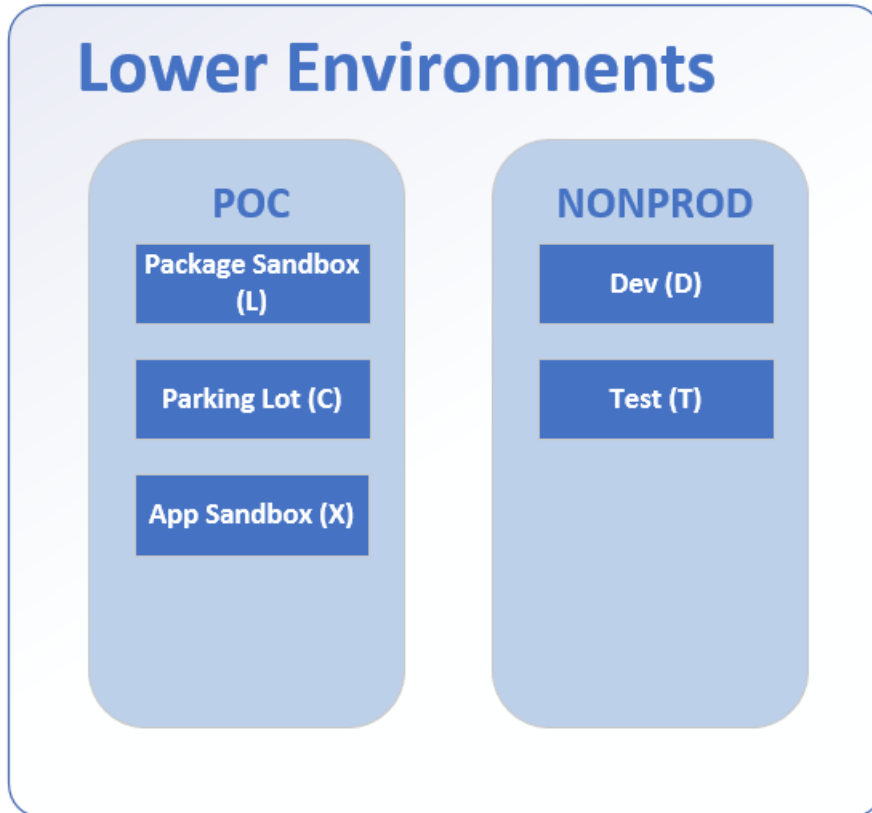
500 series – CORE

900 series – POC

Organizing Your Environment

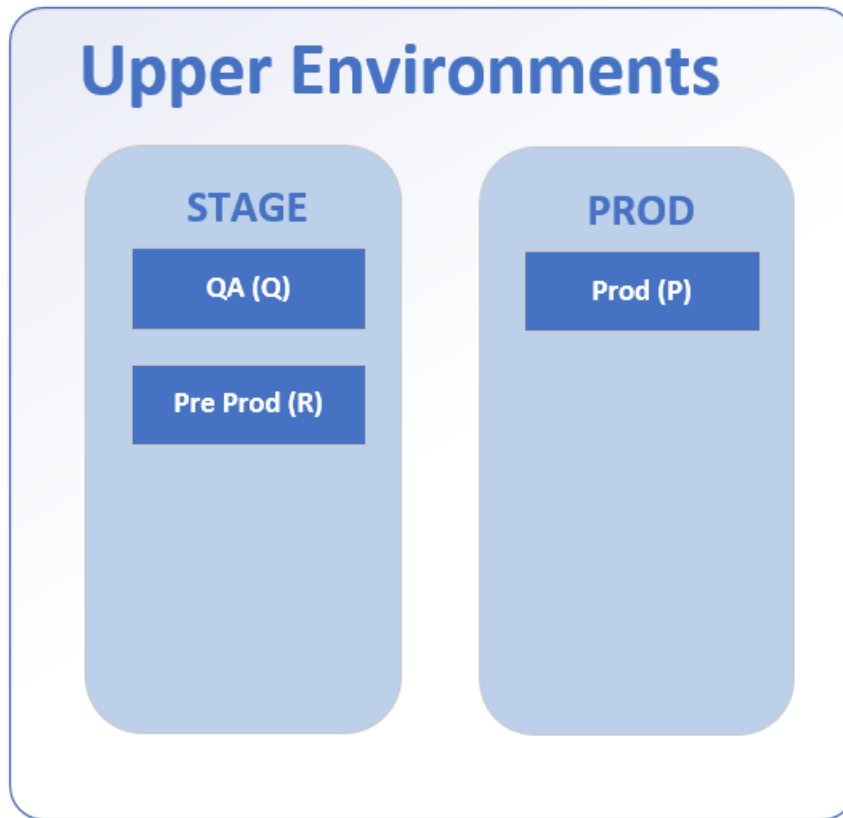


Organizing Your Environment



- Virtual Servers
- Single Instance
- No LTM Configurations
- Simple, yet purpose driven

Organizing Your Environment



- Physical or Virtual Servers?
- Multi-Instance or Active/Active
- LTM Configurations, Shared Disks, Cluster management, etc.
- Want STAGE to be as close to PROD in configurations as possible

Organizing Your Environment

Package and Parking Lot Sandboxes

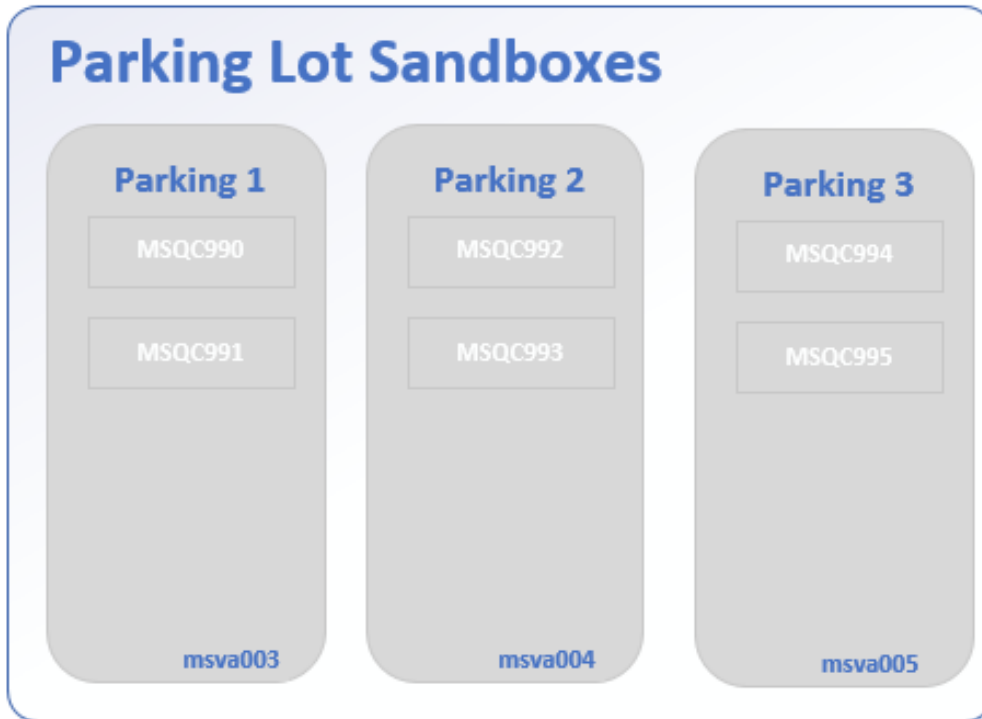


Organizing Your Environment



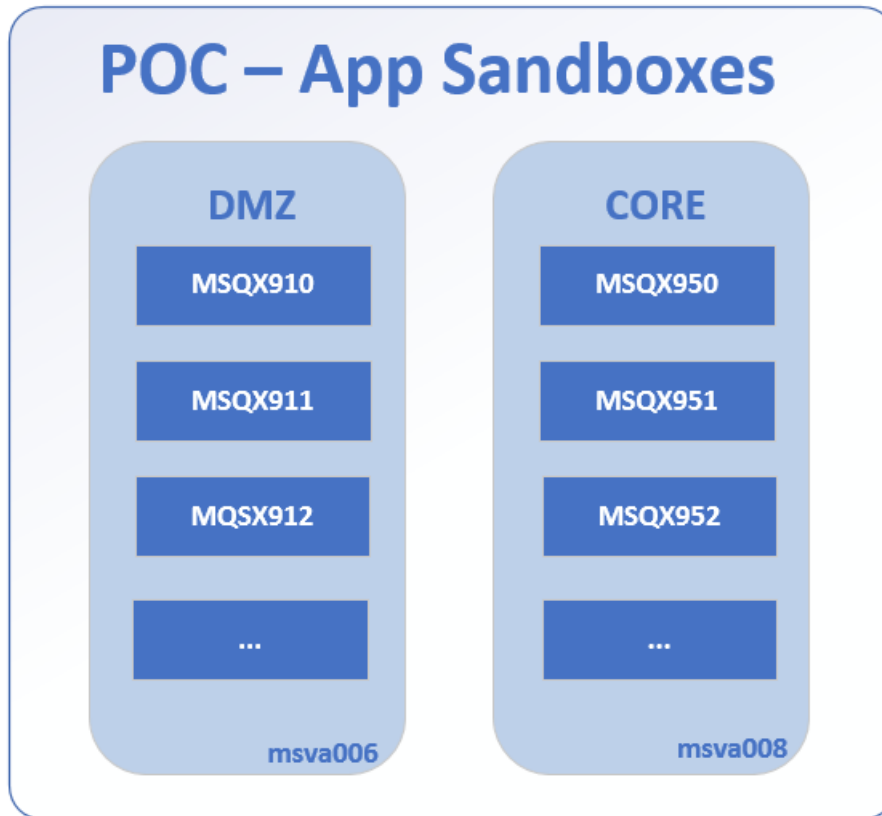
- Sandbox for testing out automation scripts for server and application buildout
- Internal Team Use

Organizing Your Environment



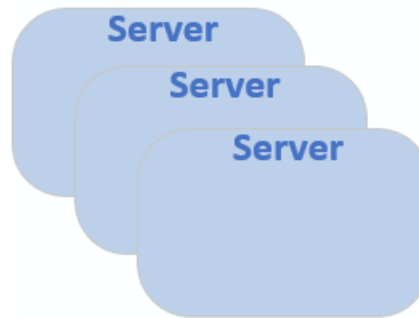
- **Multi-Purpose Sandbox**
 - Try out new CD Features
- Try out a new LTS Version
- You'd be surprised down the road finding a use for them
- Could have the servers defined and turned off until needed. Then build MQ Servers when ready.
- **Internal Team Use**

Organizing Your Environment



- **POC - App Sandbox**
 - Not Connected to MQ Cluster or other QMGRs
 - Provided as a service to App Teams if they want to “test drive” a new v9.x QMGR but not ready to commit to a migration
 - One example was to test connectivity to DynaTrace
- **App Team Use**

2.3 Server Build Scenario



Server Build Scenario

Server Build Process (In a Nutshell)

Request New Servers



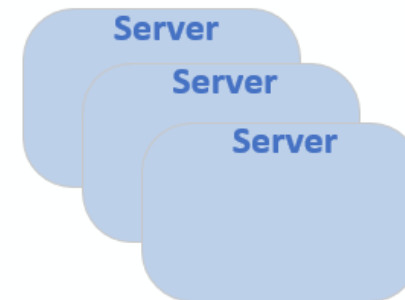
Service Catalog Request



Assign Task(s)



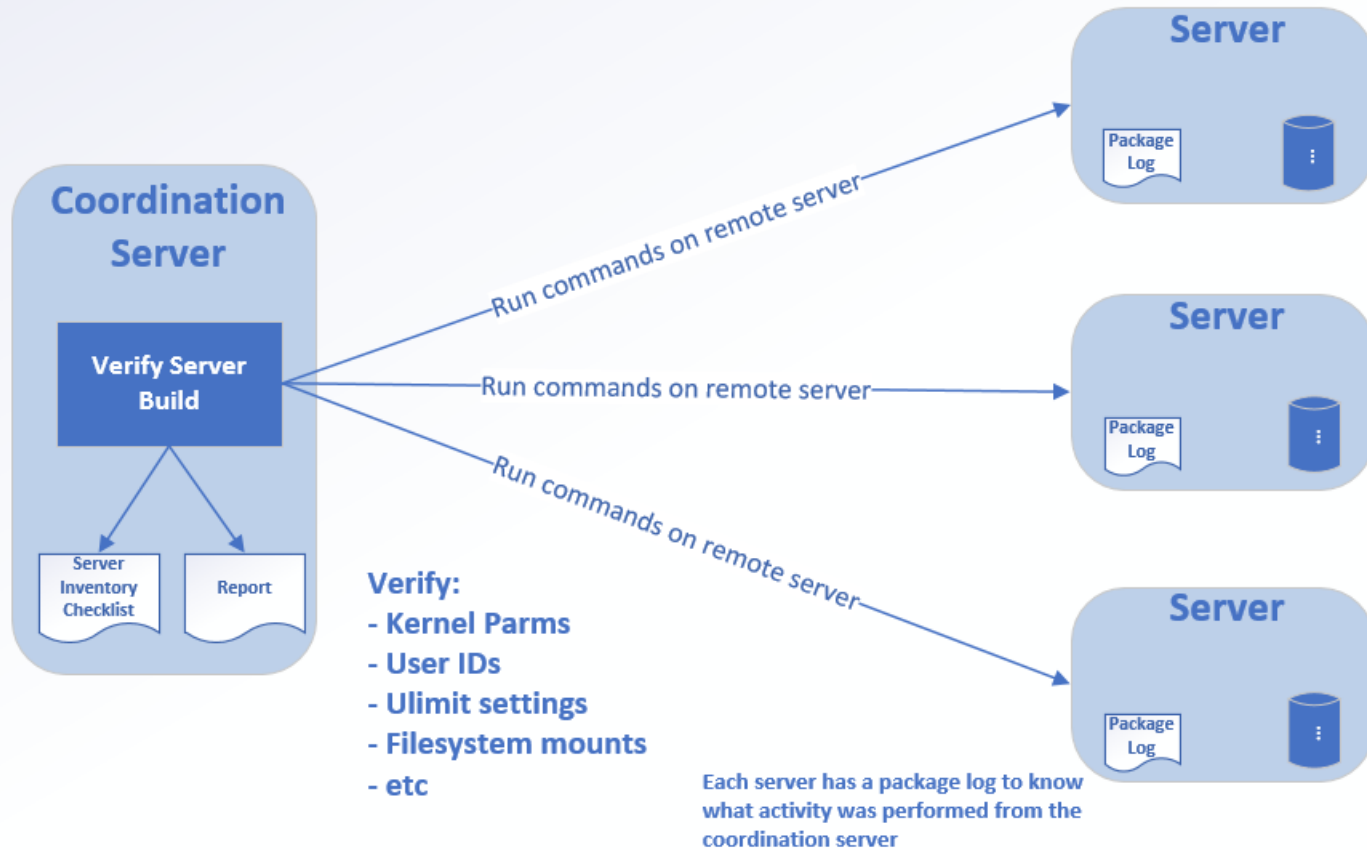
Poof! (They Exist)



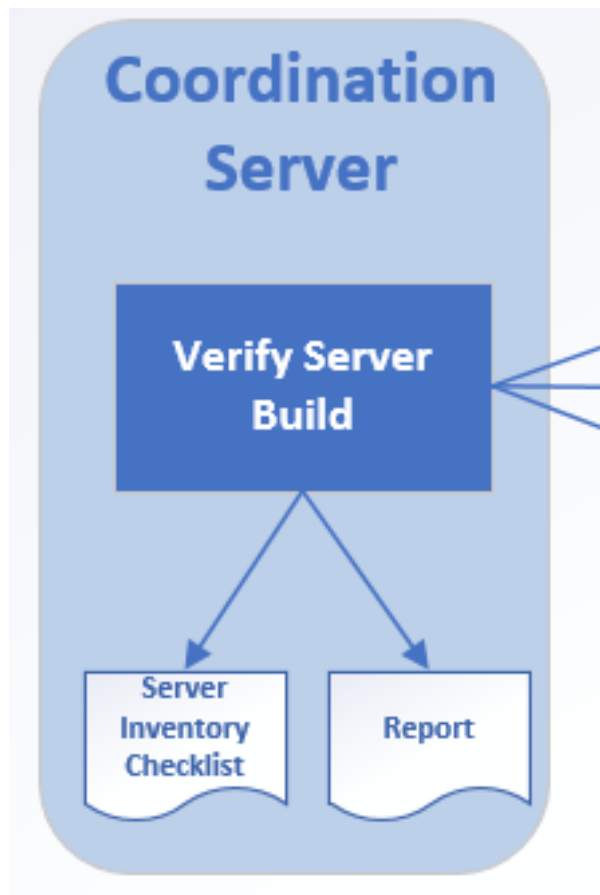
How do we know
they're correct?!

Server Build Scenario

Initial Server Verify Process (In a Nutshell)

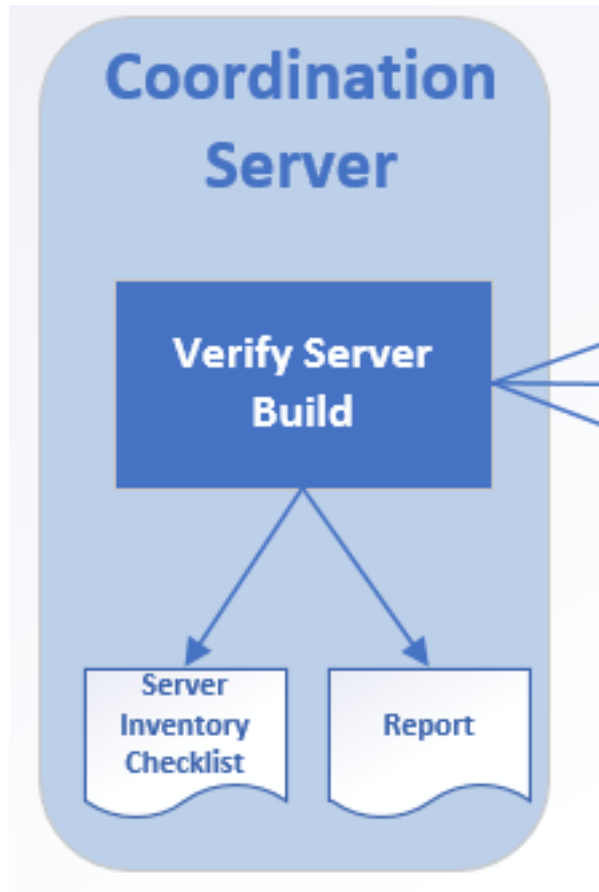


Server Build Scenario



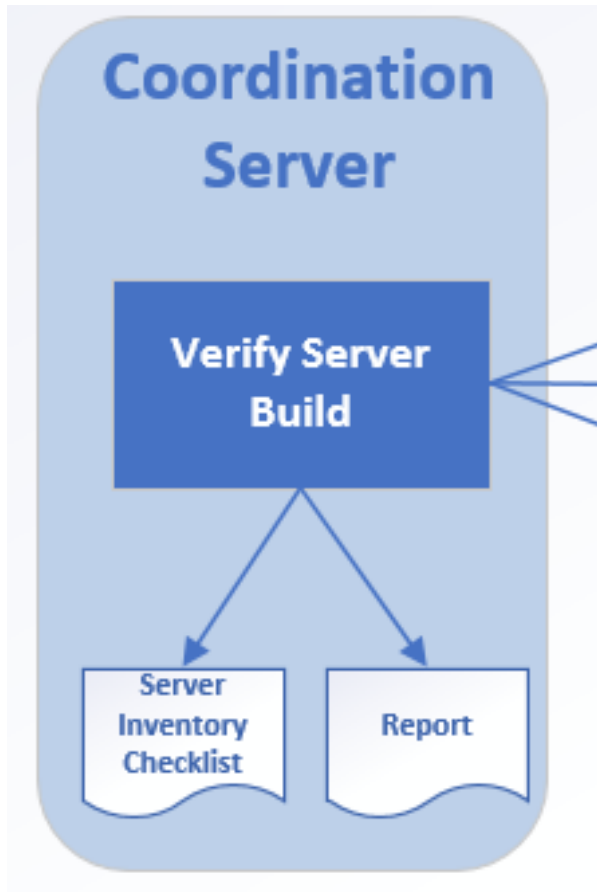
- Verify Server Build a korn script, an Ansible playbook, Chef cookbook, etc.
- Report provides details about the verify process
- Server Inventory Checklist, at it's fundamental level, is to keep track of activity on an MQ Server's progress

Server Build Scenario



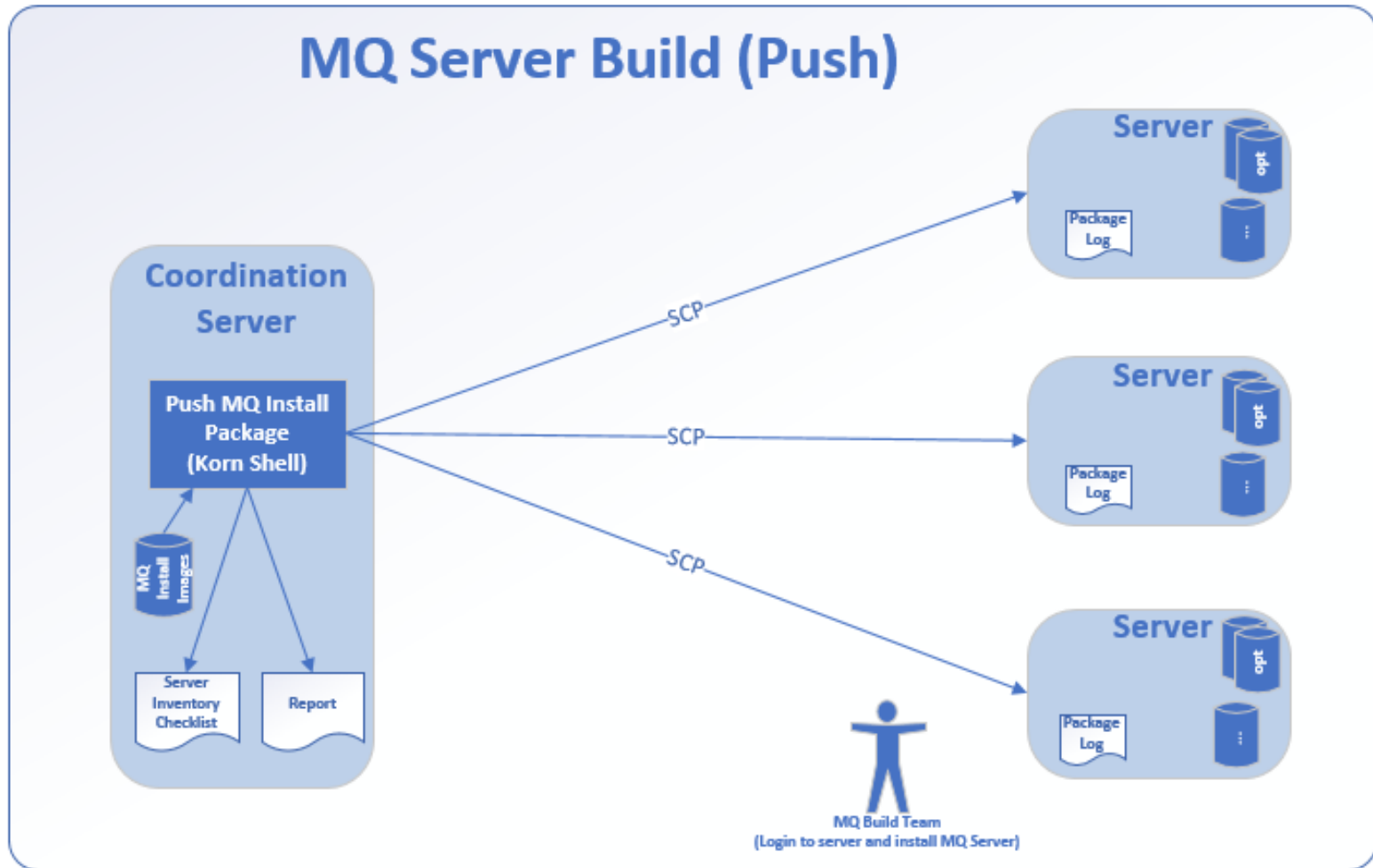
- With economies of scale, you want to reduce, if not eliminate, the need for a human to physically login to an MQ Server to perform tasks
- Building a server is a workflow in concept. Other teams work on their pieces and handoff to the MQ Team, but you'll need to verify

Server Build Scenario

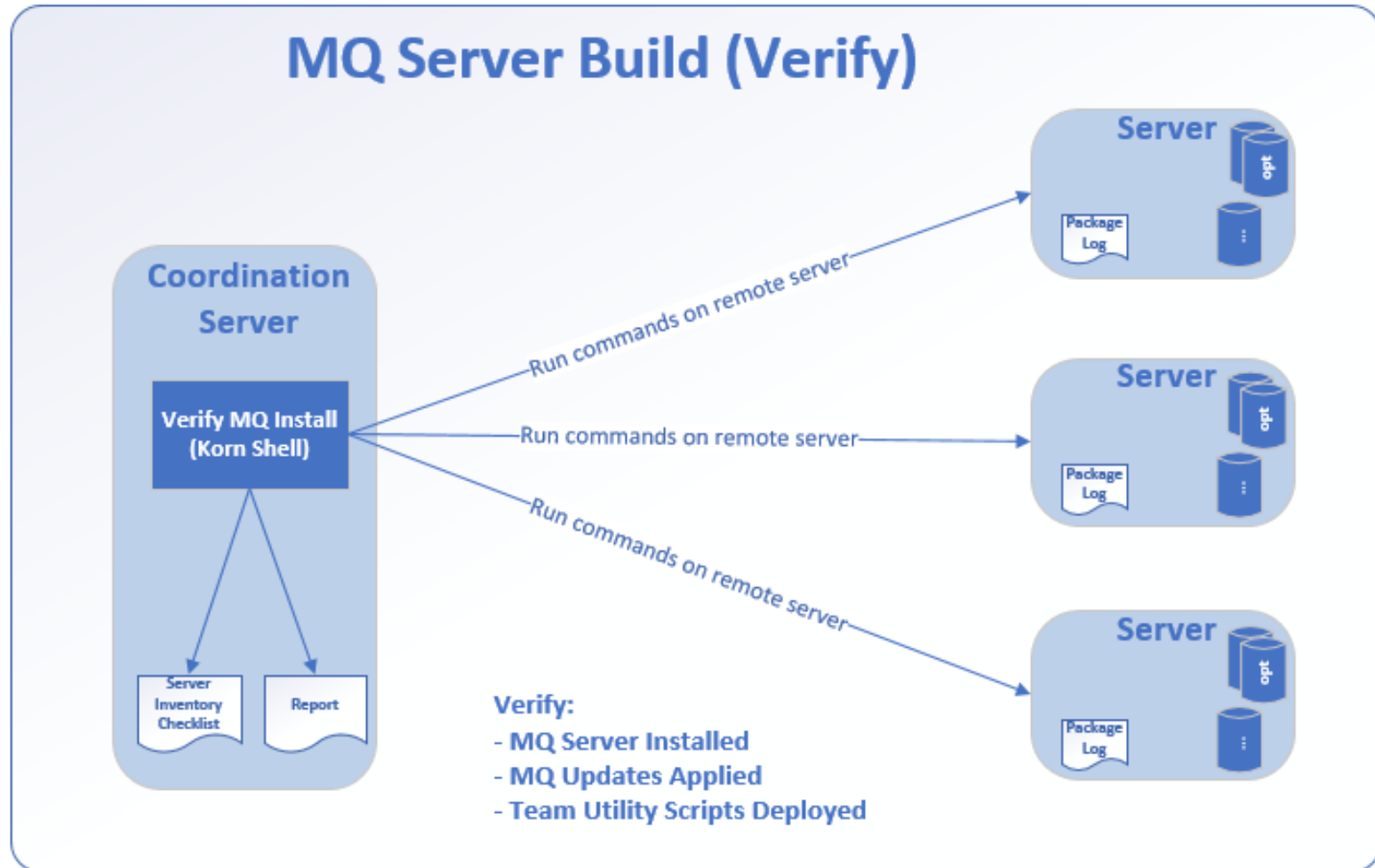


- Tasks that can be performed within the MQ Team without outside help should follow small, manageable push/verify routines
- Breaking into smaller tasks to automate allows team members to work on individual pieces of the overall workflow
- Work out patterns for push/verify routines, document, assign tasks through your Kanban board

Server Build Scenario

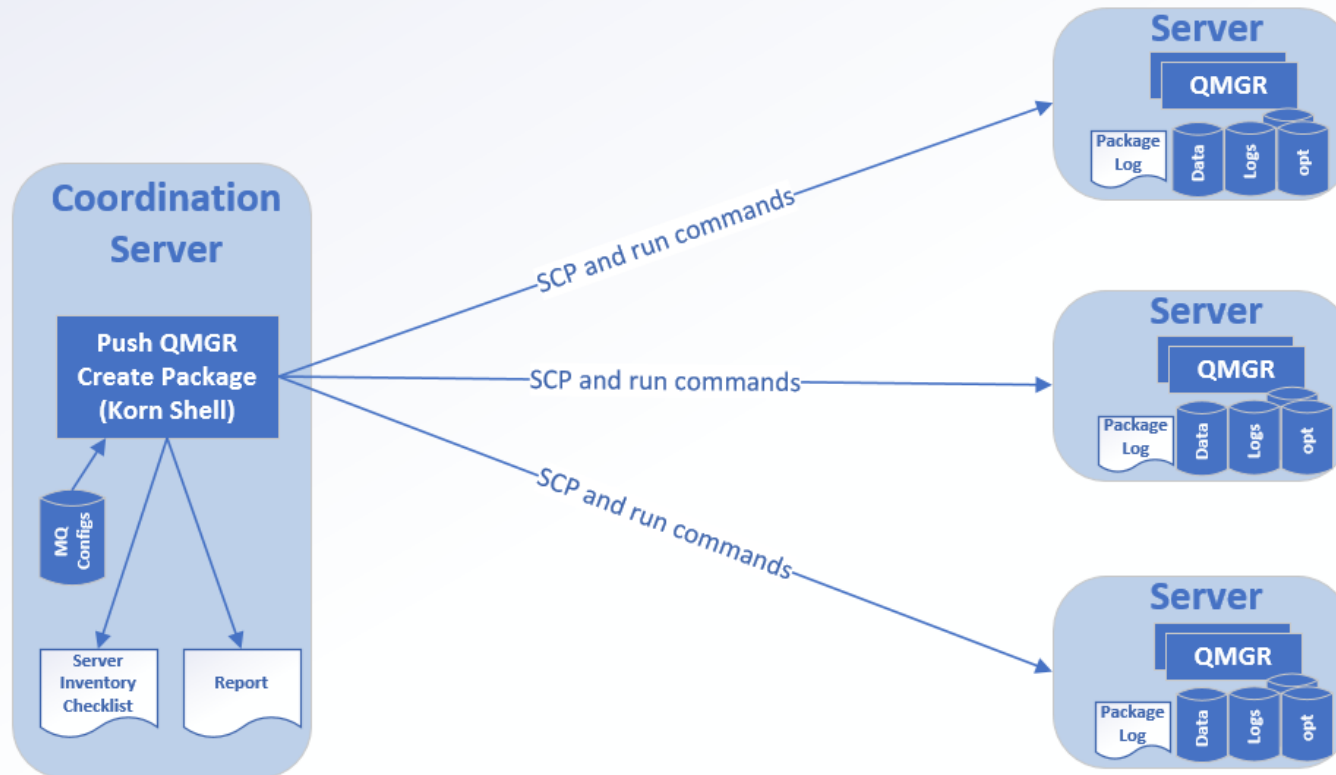


Server Build Scenario

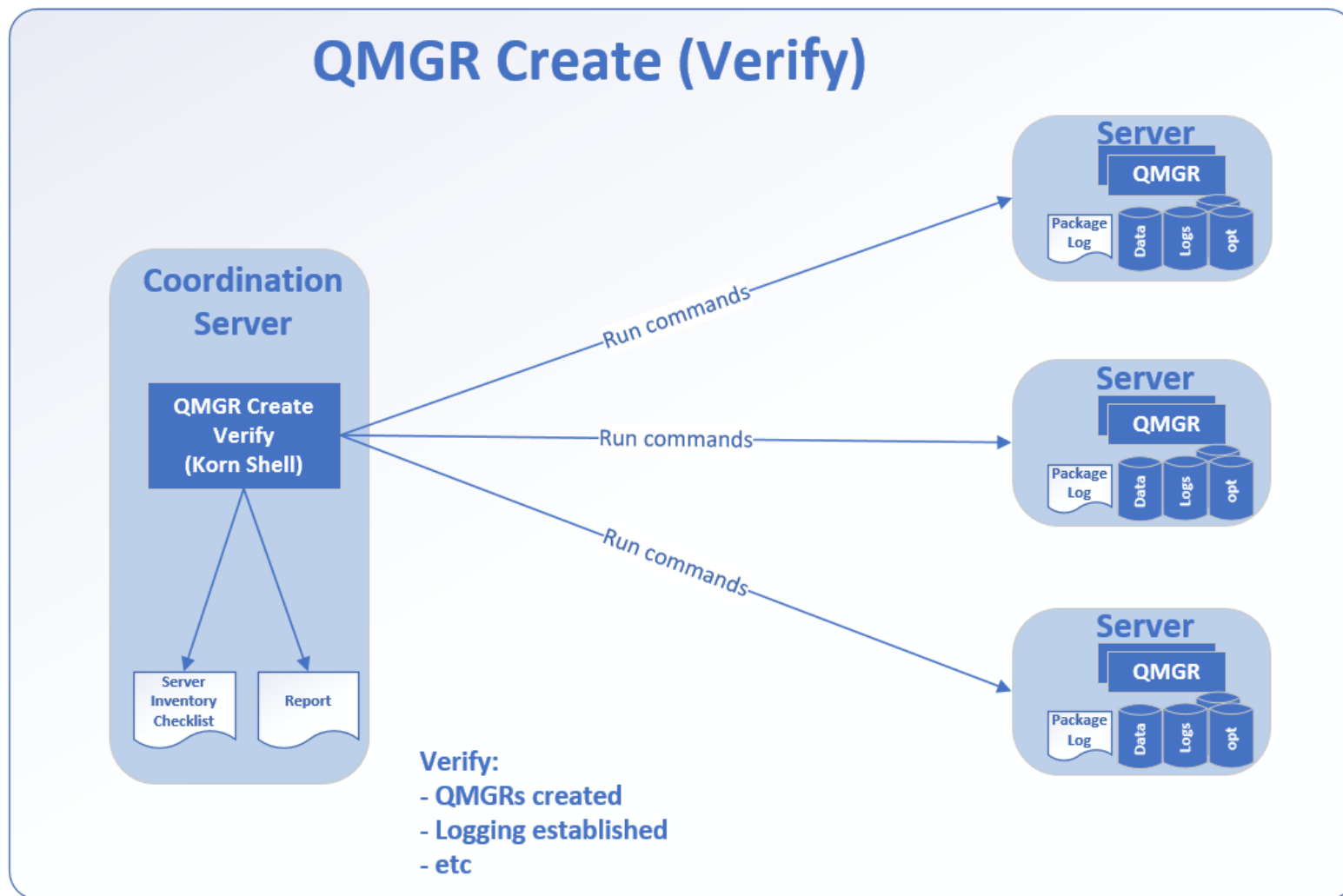


Server Build Scenario

QMGR Create (Push)

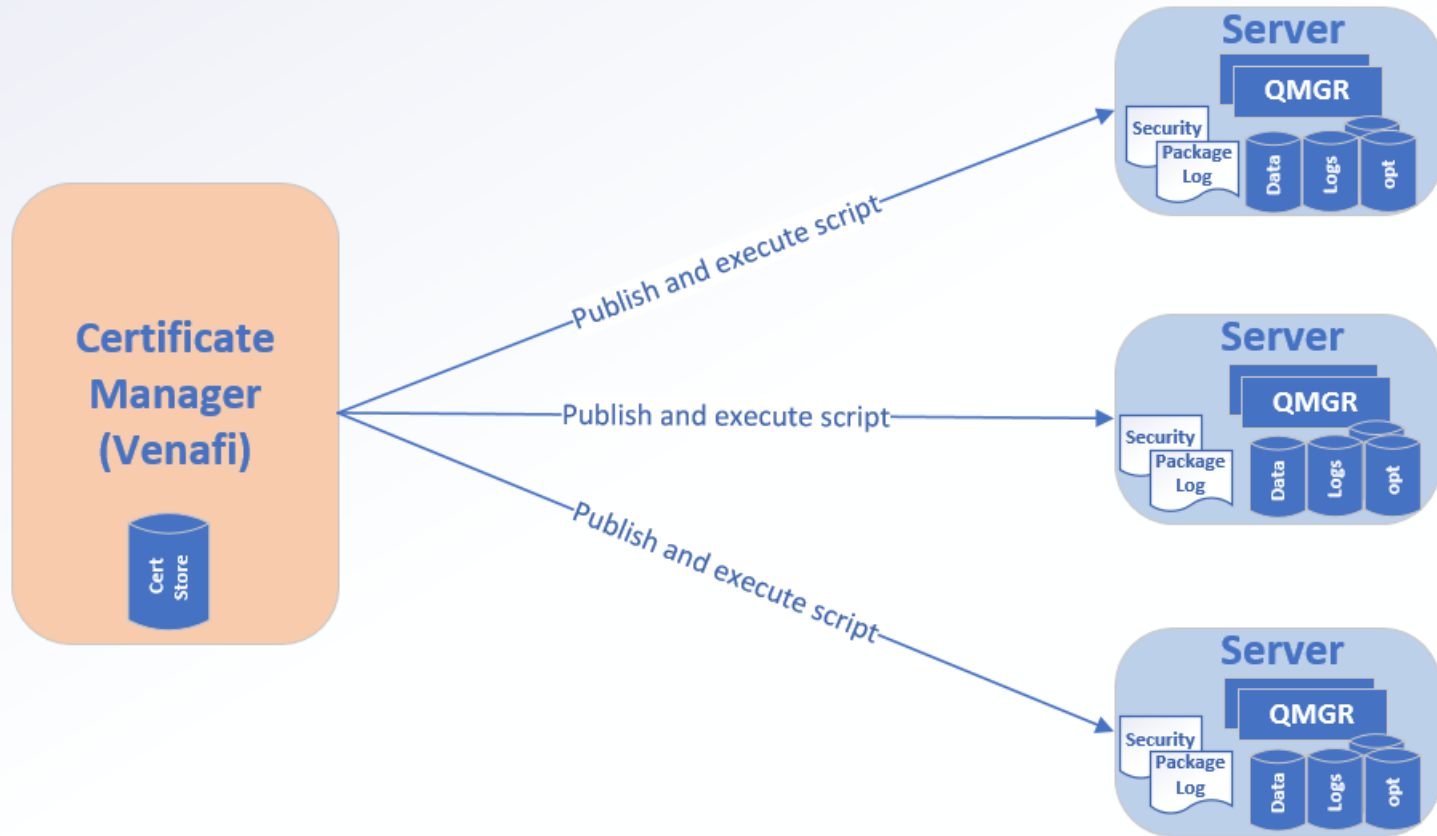


Server Build Scenario



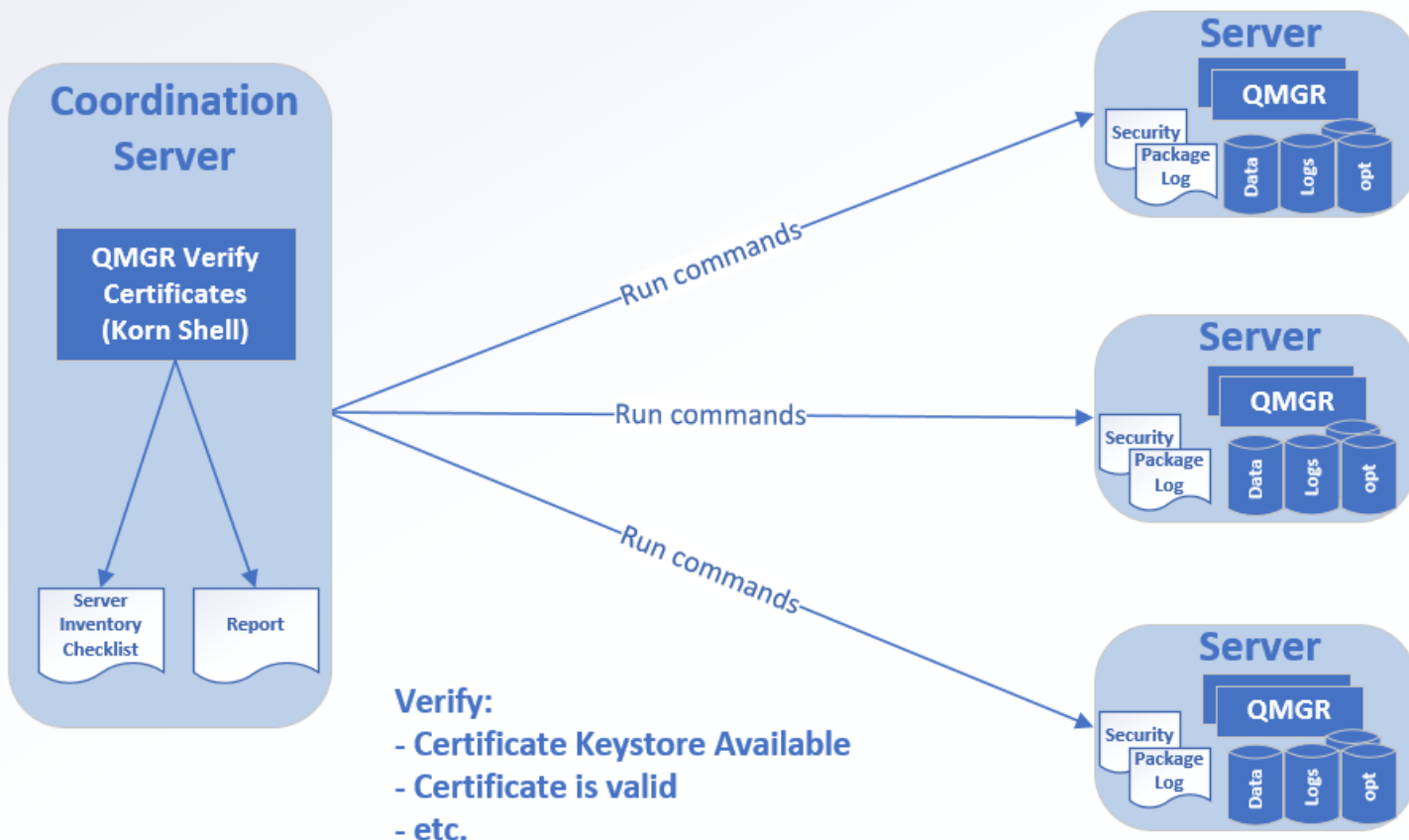
Server Build Scenario

QMGR Certificate (Push)

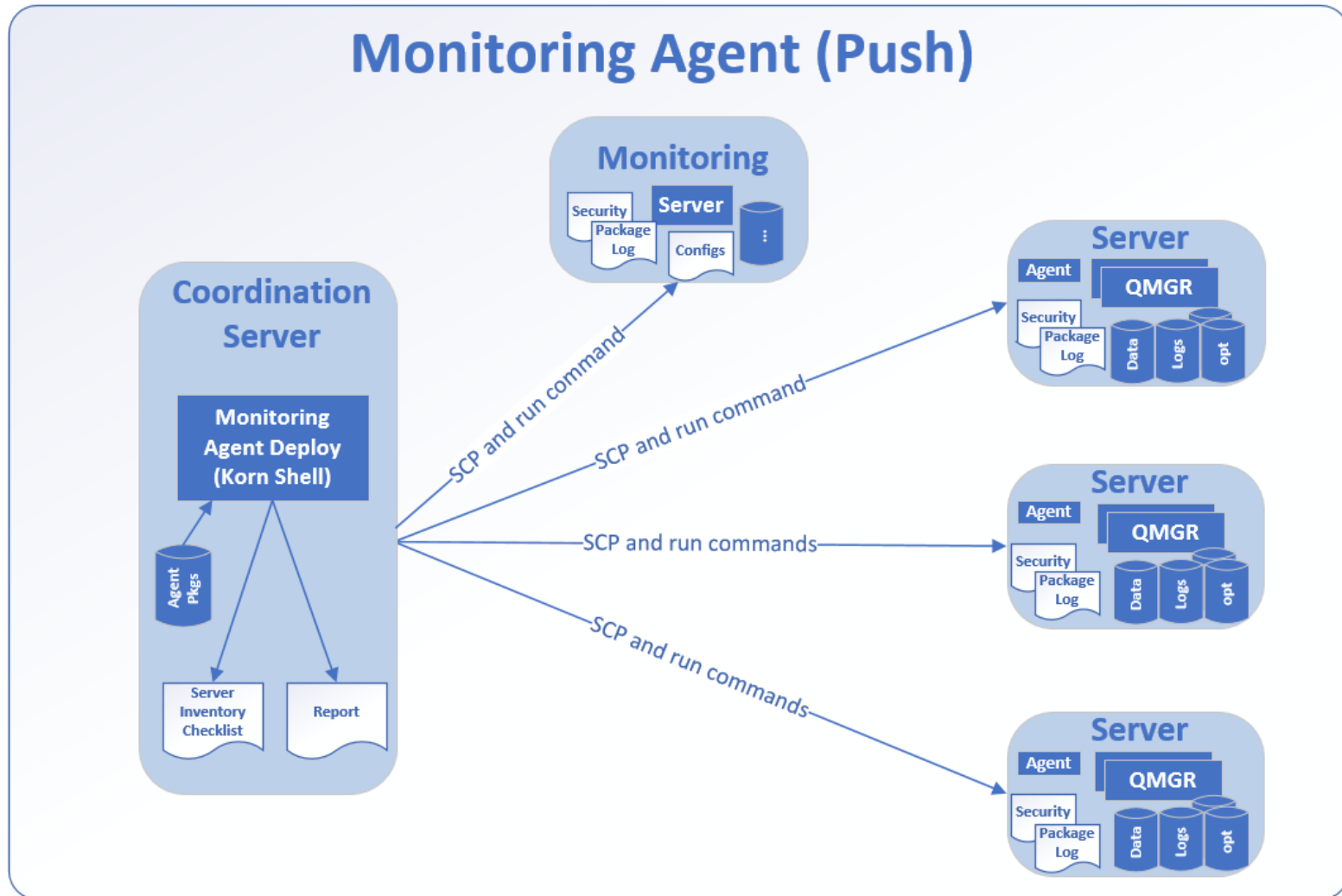


Server Build Scenario

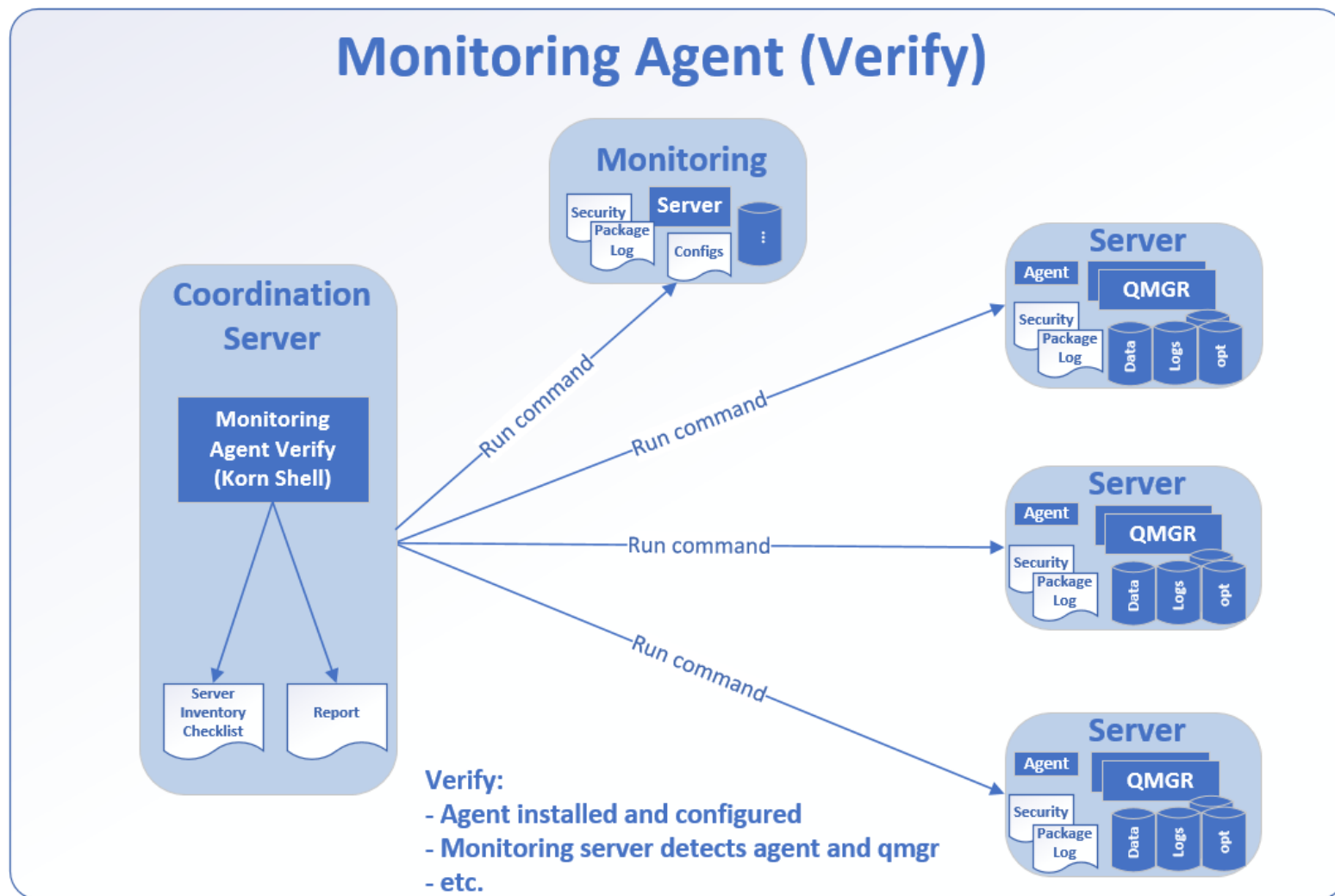
QMGR Certificate (Verify)



Server Build Scenario



Server Build Scenario



Server Build Scenario

Server	Kernel Params	User IDs	Ulimit Settings	Filesystems	Job Scheduler	MQ Server	MQ Fixpack	Team Scripts	QMGR	Logging	Certificate	Keystore	Monitoring Agent
msva002	x	x	x	x	x	x	x	x	x	x	x	x	x
msva003	x	x	x	x	x	x	x	x	x	x	x	x	x
msva004	x	x	x	x	x	x	x	x	x	x			
msva005	x	x	x	x	x	x	x	x					
msva006	x	x	x	x	x	x	x	x	x	x	x	x	x
msva007	x	x	x	x	x	x							

Server Inventory Checklist:

Through reporting, you can identify what state each server is during a server build process.

It's great because putting the extra effort to automate as much as possible ensures each server follows the exact process and reduces the chance of server “configuration drift”

Questions & Answers

